Smoothing Outlier Scores Is All You Need to Improve Outlier Detectors

Jiawei Yang*, *member, IEEE*, Susanto Rahardja*, *Fellow, IEEE*, Pasi Fränti, *senior member, IEEE*

***Abstract*—We hypothesize that *similar objects should have similar outlier scores*. To the best of our knowledge, all existing outlier detectors calculate the outlier score for each object independently regardless of the outlier scores of the other objects. Therefore, they do not guarantee that similar objects have similar outlier scores. To verify our proposed hypothesis, we propose an outlier score post-processing technique for outlier detectors, called neighborhood averaging (NA) for neighborhood smoothing in outlier score space. It pays attention to objects and their neighbors and guarantees them to have more similar outlier scores than their original scores. Given an object and its outlier score from any outlier detector, NA modifies its outlier score by combining it with its k nearest neighbors' scores. We demonstrate the effectivity of NA by using the well-known k nearest neighbors (k-NN). Experimental results show that NA improves all 10 tested baseline detectors by 13% on average relative to the original results (from 0.70 to 0.79 AUC) evaluated on nine real-world datasets. Moreover, deep-learning-based detectors and even outlier detectors that are already based on k-NN are also improved. The experiments also show that in some applications, the choice of detector is no more significant when detectors are jointly used with NA. This may pose a challenge to the generally considered idea that the data model is the most important factor. We open our code on www.outlierNet.com for reproducibility.**

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

I. Introduction

OUTLIERS are objects that significantly deviate from other objects. Outliers can indicate useful information, which can be applied in applications such as fraud detection [1], [2], abnormal time series [3], [4], and traffic patterns [5], [6]. Outliers can also be harmful because they are generally unwanted, can be considered errors, and may bias statistical analysis for applications like clustering [7], [8]. Recently, outlier detection has also been applied to manufacturing data [9] and industrial applications [10]. For these reasons, outliers need to be detected.

Most outlier detectors calculate the so-called *outlier score* for each object independently and then threshold the scores that deviate significantly from the others and label them as outliers [11]. To improve the results of baseline outlier detectors, *ensemble techniques* have been developed to combine the outcomes of multiple detectors to obtain a more accurate detector [12], [13]. An example is the *average ensemble* [1], which calculates the average outlier score from multiple baseline detectors. However, the existing ensemble techniques merely use more detectors but do not attempt to ensemble outlier scores of neighboring objects. Their success is also bounded by the reliability of the baseline detectors.

The outlier score is a fundamental concept in all score-based outlier detectors. All outlier detectors assume that outlier objects should have significantly higher or lower outlier scores [1]. Except for that, no attention has been paid to the relationship between objects and their outlier scores. Because outlier objects are directly decided by their outlier scores, it is vital to understand their relationship. In this paper, we address this problem.

In Figure 1, all detectors successfully assign significantly higher scores to the outlier eggs (red triangles) but cannot guide the selection of the best detectors. We can see that egg A is distinctive and has the highest score. Detector 2 and Detector 3 are therefore better than Detector 1. Similarly, because eggs C, D, E, and F have the same color and size, they should have the same outlier scores. In this case, Detector 3 is better than Detector 2. Therefore, we can conclude that Detector 3 is the best among the three by comparing the similarities between objects' features and their outlier scores.

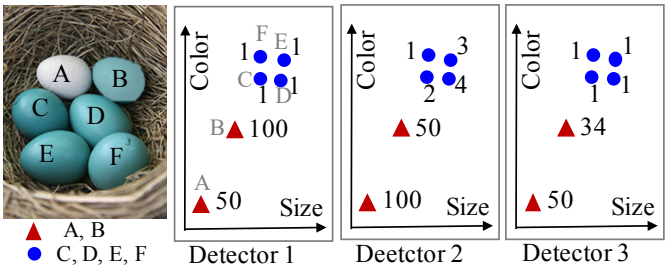


Fig. 1. Outlier scores are given by three detectors on the task of detecting outlier eggs from a Robin bird. The results of Detector 3 can be obtained from the results of Detector 2 using the proposed method as shown in Fig. 7.

Based on the case in Figure 1, we conclude that *similar objects should have similar outlier scores*. Although this could be seen as obvious, none of the state-of-the-art outlier detectors uses this. Many detectors simply make use of the objects' neighborhood in the process of producing outlier

* Corresponding author

This work was supported in part by the National Key Research and Development Program of China (STI 2030—Major Projects, Grant No. 2021ZD0201502), in part by the Oversea Expertise Introduction Project for Discipline Innovation, 111 Project, under Grant B18041, and in part by the National Science Foundation for Distinguished Young Scholars of China under Grant 62201470.

Jiawei Yang and Susanto Rahardja are with Northwestern Polytechnical University. Susanto Rahardja is also with the Singapore Institute of Technology. Pasi Fränti is with University of Eastern Finland.

{jiaweiyang,susantorahardja}@ieee.org; pasi.franti@uef.fi

scores (especially all k-NN-based detectors). However, they do not consider the relationship between objects' features and their outlier scores. For example, egg B (the red triangle in the middle) in Figure 1 is much more similar to other normal eggs C, D, E, and F compared to egg A. It should therefore have a lower score than egg A.

To address this problem, we propose a novel *neighborhood averaging* (NA) technique for neighborhood smoothing in outlier score space. It post-processes the outlier score of each object provided by any existing outlier detector by averaging it with the scores of its neighbors. In other words, if an object is an outlier, it is more likely that its near neighbors are also outliers. In this case, the predicted score is enhanced. On the contrary, if the neighboring objects have low outlier scores (predicted as normalities), the score of the object is also reduced accordingly.

The beauty of NA is that it can serve as an additional and independent post-processing technique that can be used after any existing detectors. It is different from ensemble techniques because rather than operating the results of multiple detectors of a single object, NA operates the results of multiple objects of a single detector as shown in Figure 2. NA is conceptually and fundamentally different from ensemble techniques. It is also complementary to the ensembles, and these two approaches can be used jointly. While ensembles cannot ensure similar objects have similar outlier scores, NA can achieve this.

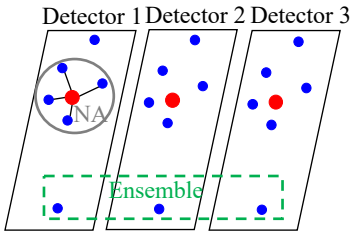


Fig. 2. Difference between NA and ensembles. Ensembles use multiple detectors' prediction of the same object (on the bottom), while NA uses a single detector's prediction of the different (neighboring) objects (with a gray background).

Figure 3 demonstrates all the combinations that can be constructed from NA and the existing outlier detection methods, including ensemble techniques. On the top, we have the typical situation where dataset X is input into an outlier detector, which produces scores that are further processed by a threshold component to determine outliers. The second case is the multi-detector ensemble where the dataset is input into two outlier detectors to produce two separate scores. The scores are then combined by the ensemble component before they are processed by the threshold component to determine the outliers. The third case is the proposed NA where the dataset is input into an outlier detector, after which the scores are averaged before they are processed by the threshold component. The last case is a combination of the multiple-detector ensemble and NA, where two outlier detectors produce scores that are first combined by the ensemble and then post-processed by NA.

To summarize this paper's contribution, (1) we assume that similar objects in feature space should have similar outlier scores. (2) We propose NA based on k-NN to post-process the existing outlier scores to produce more reliable and consistent scores for neighborhood smoothing in outlier score space. While there are already many k-NN-based methods, they all operate in the feature space. In contrast, NA operates in the score space by modifying existing scores without any additional information besides the neighborhood graph defined in the feature space. The method is not limited to geographical data [14] or any other single application, but it can be applied in any application domain. It can improve any existing score-based outlier detectors or ensemble techniques, and it is not limited to use with k-NN-based outlier detectors. (3) We perform comprehensive experiments showing NA's superiority when jointly used with existing outlier detection techniques.

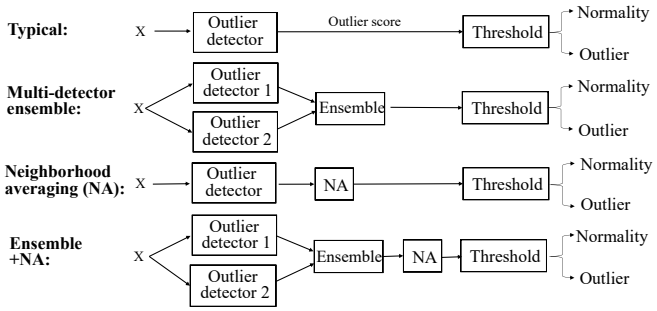


Fig. 3. Outlier detection process.

We organize this paper as follows. In Section II, we recall several state-of-the-art outlier detectors from several categories. They later are used as our baseline detectors. In Section III, we introduce the proposed hypothesis and NA. The experimental setup is described in Section IV, and the results are shown in Section V. In Section VI, we describe our conclusions.

II. OUTLIER DETECTORS

We next review the existing outlier detection methods. They all analyze the relationship between the objects globally or locally and calculate an independent outlier score to conclude whether an object is an outlier. NA can be applied to all of these as an outlier score post-processing technique, and to the best of our knowledge, there is no similar technique in the literature. All introduced detectors consider only the relationship between objects' features and, unlike our technique, they do not operate on the outlier scores.

By constructing the *reference set* [1] for the calculation of outlier scores, outlier detectors can be grouped into global detectors and local detectors. Global detectors use all objects and local detectors use only a small subset of objects, such as k-NN, in the dataset as a reference set. We next review 12 well-known and state-of-the-art outlier detectors including deep-learning-based detectors.

In distance-based outlier detectors [15]–[17], outlier objects essentially should be located far away from other objects. The detector proposed by Ramaswamy *et al.* [15] computes the distance between an object and its k^{th} nearest neighbor as the outlier score. This detector is referred to as KNN [15]. A

variant that evaluates the average distance to its all k neighbors was proposed by Hautamäki *et al.* [16]. The method proposed by Shekhar *et al.* [14] calculates the distance to the average of its k-NN. It uses spatial features to determine the neighbors and the other features for the outlier detection.

Instead of considering the distance, the detector proposed by Knorr *et al.* [17] counts the number of objects within a predefined distance threshold to the object. The count is used as the outlier score. Outlier detection using indegree of nodes (ODIN) proposed by Hautamäki *et al.* [16] is also based on the k-NN graph. It uses the number of being other objects' neighbors as the outlier score.

Reverse unreachability (NC, as defined by Li *et al.* [18]), is a detector based on representation. A given object is represented by k-NN with a weight matrix corresponding to the contribution from each neighbor. The negative weights carry information on the possibility of being outliers. The occurrence of negative weights is used as the outlier score.

Mean-shift outlier detection (MOD) [7], [8], [19] replaces an object with its k-NN's mean. This process is repeated three times. The distance between the original object and the modified object is the outlier score. This approach works well, especially when a dataset contains a large number of outliers [7].

In density-based detectors [20], [21], outlier objects have considerably lower densities than their neighbors. Local outlier factor (LOF) [18] evaluates the density of an object relative to that of its k-NN as the outlier score. In [22], LOF was reported to be the best-known detector when compared to the other 12 k-NN-based detectors.

The minimum covariance determinant (MCD) [23] is based on statistical analysis and is a robust estimator for evaluating the mean and covariance matrix. It finds 50% of objects with a covariance matrix having the smallest determinant. It then uses the difference from an object to the center of the objects as the outlier score.

Isolation-based anomaly detection (IFOREST, as defined by Liu *et al.* [24]) builds trees over the dataset. It recursively separates the objects into two parts with a threshold randomly selected from each dimension. To remove the bias of randomness, it repeats the process several times. The average number of splits to isolate an object from other objects is its outlier score. An improved version of IFOREST can be found in [25].

Support vector machine (SVM) has been widely applied to pattern recognition tasks. One class support vector machine (OCSVM) [26] treats the objects as training data and creates a one-class model. The distance to the trained model is then used as the outlier score.

Principal component analysis (PCA) is an established data-mining technique. PCA can extract the principal structure of the data. The principal-component-analysis-based outlier detection method (PCAD) [27] reconstructs objects using the eigenvectors with reconstruction errors. The normalized errors are outlier scores.

Angle-based outlier detection (ABOD) [28] calculates the angles between objects. The variance of these angles is used as the outlier score. It was viewed as overcoming dimensionality better than distance-based measures in [28].

Multiple-objective generative-adversarial active learning (MO-GAAL) [29] is proposed to overcome the sparsity of data in high-dimensional space by generating additional data objects. MO-GAAL first trains a neural network to classify the generative and real-data objects. The outlier score is calculated as the possibility of the object being real.

Copula-based outlier detector (COPOD) [30], [31] predicts the tail probabilities of each object by constructing an empirical copula. The probability is used as the outlier score.

To sum up, the above-mentioned detectors can be divided into four categories: proximity-based detectors (KNN, ODIN, NC, MOD, LOF, and ABOD), statistics-based detectors (MCD and PCA), learning-based detectors (MO-GAA and SVM), and ensemble-based detectors (IFORES and COPOD). Regardless of the categories of detectors, outlier scores for different objects have been generated independently without considering the scores of other objects. This will lead to inconsistent scores for similar objects, in which case NA will be needed to smooth these inconsistent scores to improve detectors.

III. Methodology

In this section, we present the general framework of NA. In general, outlier detectors utilize different assumptions to produce outlier scores, such as distance or density. However, NA does not set any requirements but assumes that similar objects in the feature space should have similar outlier scores.

A. General averaging framework

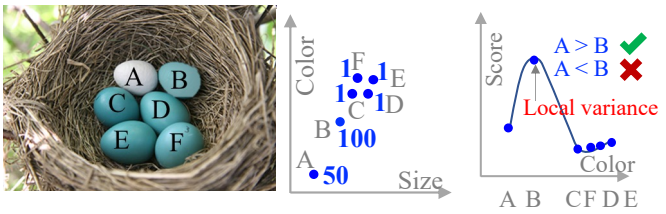


Fig. 4. Define local variance in outlier scores: relative outlier scores do not match the relative degree of being outliers.

The example in Figure 1 shows that *similar objects should have similar outlier scores*. Although Detector 1 can find the two outliers (with a proper threshold), by plotting the outlier scores in Figure 4, we can see there is a local peak in the distribution of the outlier scores, which does not match reality. Figure 5 shows that the local peak will cause either a false positive or a false negative regardless of which threshold value is selected. It is therefore necessary to remove the local peak.

In a recommendation system [32], a related hypothesis for collaborative filtering techniques states that *similar users must/should have similar preferences*. Both of these hypotheses rely on defining the similarity of the objects in the feature space. However, there is one important difference between them. While collaborative filtering does not involve any outlier score calculations, the definition of the outlier score is the key to outlier detection.

Figure 6 shows three types of similar objects. Various distance metrics can be employed to define similar objects.

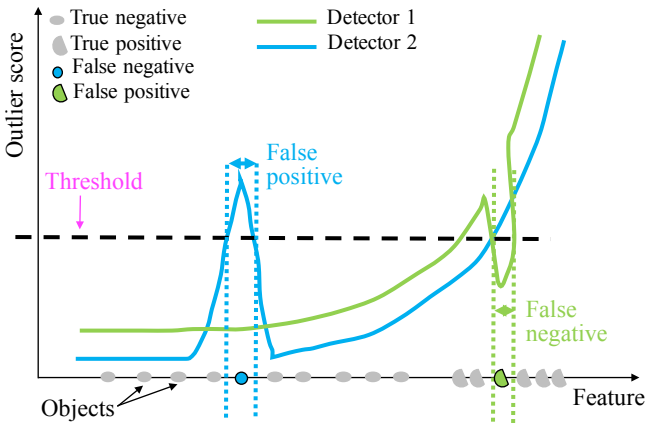


Fig. 5. Visualization of how the local variance (local peak) affects the accuracy of outlier detection. The blue line and green line have local variances. We can see that no matter how we adjust the threshold value, the local variance affects the accuracy by causing either a false positive or a false negative.

For tabular data, the Euclidean distance, Manhattan distance, or Cosine distance can be utilized. Set data can be evaluated using the Jaccard distance, while string data can be assessed using the Hamming distance or Edit distance. Spatial data can be measured using the Haversine distance, and graph data can be analyzed using the Geodesic distance. In the case of a data type with multiple options for distance metrics, the selection of an optimized distance metric may be determined based on the distribution of the data and the patterns of outliers. However, due to constraints on the length of this paper, a detailed exploration of this topic will be deferred to future research. With the aid of distance metrics, similar objects can be defined as objects with a sufficiently small distance. Optionally, similar objects can be defined as objects within the same partition after performing a data partition. The accurate definition of similar objects has a direct impact on the performance of NA. In cases where the boundary between outliers and normalities is uncertain, the application of a smoothing method may have a detrimental effect on performance. To mitigate this issue, it is crucial to precisely define similar objects. This can be accomplished by either defining objects within a narrow region as similar or by considering objects that are in close proximity to each other using various distance metrics.

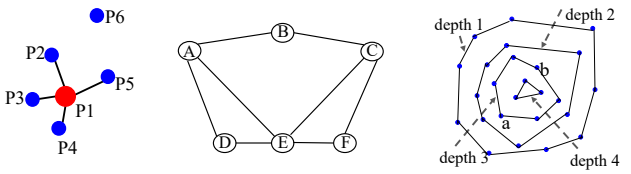


Fig. 6. Definition of the similarity of objects can be different with different data. In feature space, it can be based on the distance of objects (Left); it can be based on the nodes' common neighbors (Middle); and it can be based on which level the object is located in the structure (Right).

B. Neighborhood averaging (NA)

The proposed NA technique is simple: We take any baseline outlier detector and use it to compute the preliminary outlier score for every object. We then modify the objects' outlier scores in the neighborhood to be closer to one another to smooth the baseline outlier detectors' results. The main advantage of NA is its applicability to any existing outlier detectors or ensemble techniques. While we use k-NN defined by Euclidean distance in this paper, it should be noted that any neighborhood model can also be applied.

NA first defines an outlier score similarity function $f(\cdot)$ as Equation 1.

$$f(\theta_i) = \sum_j W_j(\theta_i - S_j)^2, X_j \in \psi_i^k, S_j \in S \tag{1}$$

where ψ_i^k is defined as the set containing X_i and its k-NN and W_j is a weight satisfying $\sum_j W_j = 1, W_j > 0$. Then, NA finds the θ_i such that the $f(\cdot)$ can be minimized as Equation 2 and uses the θ_i found as the revised outlier score for object X_i.

$$\theta_i \leftarrow \underset{\theta_i}{\operatorname{argmin}} f(\theta_i) \tag{2}$$

To obtain the θ_i to minimize the $f(\cdot)$, NA gets $\theta_i = \frac{\sum_j W_j S_j}{\sum_j W_j}$ after solving the equation $\frac{\partial f}{\partial \theta_i} = 0$. The weight W_j is influenced by two factors: the similarity of objects and the reliability of the scores. When the scores are more reliable and the objects are more similar, the weight W_j increases. Therefore, it is not recommended to use the distances (or normalized within k-NN) between objects as the weight values directly. To simplify the solution, we can set $W_j = W_p$ for any $X_j, X_p \in \psi_i^k$. Finally, θ_i can be calculated as Equation 3. The solution with the optimal weights can be future work as validating the concept of NA is more important than focusing on developing a more sophisticated solution.

$$\theta_i = \frac{\sum_j S_j}{k+1}, X_j \in \psi_i^k, S_j \in S \tag{3}$$

Algorithm 1 NA(X, S, k)

Input : Dataset X, Raw outlier scores S, Neighborhood size k
Output: Revised outlier scores θ
foreach $X_i \in X$ **do**
  calculate θ_i via Equation 3;
end

Algorithm 1 shows the pseudo-code and Figure 7 demonstrates NA's two steps. Considering the red object (object B in Figure 4), NA first searches its k-NN and then calculates the average scores of the neighbors. As a result, the peak in the outlier scores in Figure 4 has been removed. The visualization examples with and without NA are shown in Figure 8. We can see that the LOF detector (with k = 40) falsely detects many boundary objects as outliers (cross), but it succeeds after using NA.

NA updates the outlier score of an object by the average of the scores of its neighbors. Where the object is also a neighbor of other objects, NA would be applied with multiple iterations,

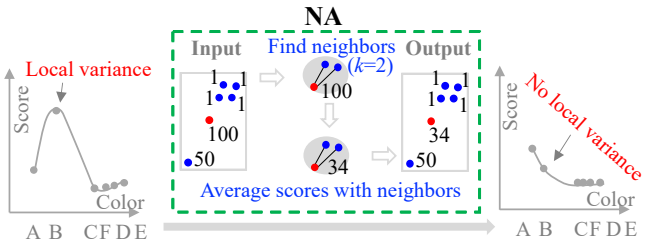


Fig. 7. Illustration of the averaging process: an object B from Figure 4 (red), and all the outlier scores. NA first finds the 2 nearest objects of B and then calculates the average score within its neighbors as the revised score: (100+1+1)/3 = 34. As a result, the local peak has been successfully removed by NA.

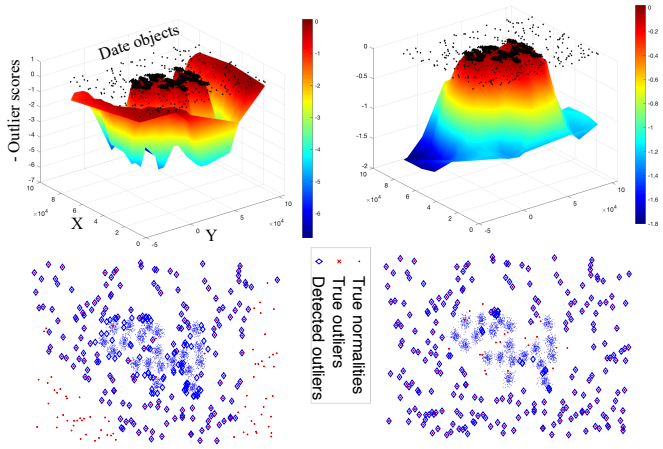


Fig. 8. Visualization of the outlier scores (Top row) and the detected outliers (Bottom row). The results at the left column and right column are given by the LOF detector with and without NA, respectively (k = 40). LOF falsely detects boundary objects as outliers (cross) as evaluated on a noisy A1 dataset [33], while NA improves the result of the LOF significantly.

and in each current iteration, only the score of the last iteration is used to revise each object's score. The iteration leading to coverage of outlier scores of all objects in the dataset should be avoided. More information about the effect and selection of the iteration can be found in Section V.B.

C. Theoretical analysis

Theoretical analysis with the bias-variance tradeoff: The idea of analyzing the unsupervised outlier ensemble using the bias-variance tradeoff, as proposed by Aggarwal [34], can be employed to analyze NA. In this analysis, we make the assumption that the ideal (ground truth) outlier score for a given object is determined by an unknown function $g(\cdot)$, which can be estimated using outlier detectors such as $h(\cdot)$. Both $g(\cdot)$ and $h(\cdot)$ produce scores that satisfy the assumption of having a zero mean and unit variance across all objects $X_i \in X$. The mean-squared error (MSE) of the detector $h(\cdot)$, calculated over all the test objects in $X_i \in X$, is defined as $MSE = \frac{1}{N}\sum_{i=1}^{N}\{g(X_i) - h(X_i)\}^2$.

After introducing a random noise term ε_i (variable) to the X_i, denoted as $\hat{X}_i = X_i + \varepsilon_i$, the score of X_i can be calculated by ensembling the outlier scores of $\hat{X}_i$ with different noise ε_i. Consequently, the expected MSE over various ε_i is represented by Equation 4, where $E[\cdot]$ represents the mathematical expectation. The first and second components of Equation 4 correspond to the (squared) *bias* and *variance*, respectively. In other words, Equation 4 is equivalent to the expression $E[MSE] = bias^2 + variance$. To minimize the value of $E[MSE]$, techniques can be developed to reduce either the bias or variance component. However, NA focuses on reducing the variance term by setting $\hat{X}_i = X_i + \varepsilon_i = X_j$, where $X_j \in \psi_i^k$ (as defined in Equation 1). Since X_i and X_j are close to each other in terms of distance, the noise $\varepsilon_i = X_i - X_j$ is also small. Consequently, the difference $E[h(\hat{X}_i)] - h(\hat{X}_i)$ is also small, leading to a reduction in the variance component.

$$
\begin{aligned}
E[MSE] &= \frac{1}{N}\sum_{i=1}^{N} E[\{g(X_i) - h(\hat{X}_i)\}^2] \\
&= \frac{1}{N}\sum_{i=1}^{N} E[\{g(X_i) - E[h(\hat{X}_i)] + E[h(\hat{X}_i)] - h(\hat{X}_i)\}^2] \\
&= \frac{1}{N}\sum_{i=1}^{N} E[\{g(X_i) - E[h(\hat{X}_i)]\}^2] \\
&+ \frac{2}{N}\sum_{i=1}^{N} \{g(X_i) - E[h(\hat{X}_i)]\}\{E[h(\hat{X}_i)] - E[h(\hat{X}_i)]\} \\
&+ \frac{1}{N}\sum_{i=1}^{N} E[\{E[h(\hat{X}_i)] - h(\hat{X}_i)\}^2] \\
&= \frac{1}{N}\sum_{i=1}^{N} E[\{g(X_i) - E[h(\hat{X}_i)]\}^2] \\
&+ \frac{1}{N}\sum_{i=1}^{N} E[\{E[h(\hat{X}_i)] - h(\hat{X}_i)\}^2]
\end{aligned}
\tag{4}
$$

Theoretical analysis with a case: It is very challenging to theoretically prove that the revised score calculated via Equation 3 can improve the outlier score reliability because the proof process has to consider all the factors that affect the outlier detection. These include the data dimensions, the data types, the number of clusters, the properties of clusters such as shape, size, and density, the outlier types, and the number of outliers. Therefore, we give a *theoretical analysis* of how the revised score calculated via Equation 3 can improve the reliability of outlier scores by analyzing an example as shown in Figure 9.

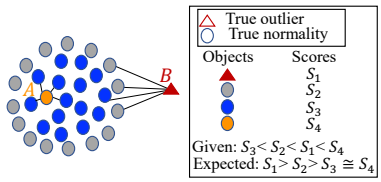


Fig. 9. An example where a normality object (point A, yellow circle) has a larger outlier score than that of an outlier object (point B, red triangle).

Figure 9 shows an example that the expected scores should satisfy $S_1 > S_2 > S_3 \cong S_4$, while $S_4 > S_1 > S_2 > S_3$ is given. Here, S_1 is the outlier score for the outlier object (point B,

red triangle), S_2 are the outlier scores for boundary objects (grey circle, normalities), S_3 are the outlier scores for insider objects (blue circle, normalities), and S_4 is the outlier score for normality object (point A, yellow circle). NA revises the scores so that the revised score θ_n for normalities and the revised score θ_o for the outlier satisfy $\theta_o > \theta_n$ as shown in Equation 5. Here, k_i^n and k_i^o are the numbers of S_i in the neighborhood of a normality object and an outlier object, respectively. If the solution for Equation 5 exists, the neighborhood of point A should not contain the point B, while the neighborhood of point B should not contain the point A, as shown in Equation 6. If the Equation 6 holds, we can get $k_1^n = 0$ and $k_4^0 = 0$. With Equation 5, we can obtain $k_2^o > k_2^n + \frac{(S_4 - S_1)}{S_2 - S_3}$ as shown in Equation 7.

Finally, the solution for $\theta_o > \theta_n$ is the k satisfying $k^{lower} < k < k^{upper}$ as shown in Equation 6 and Equation 7. Therefore, we can see that NA works when some conditions hold. First, decided by the upper bound k^{upper}, point A and point B should have enough distance so that their neighborhoods are not much overlapped. Second, decided by the lower bound k^{lower}, the outlier score difference between the point A and point B, namely the term $S_4 - S_1$, should not be significantly larger than the outlier score difference between the boundary objects and insider objects, namely the term $S_2 - S_3$. If these conditions are not met, NA may still work by requiring a larger k until more necessary boundary and insider objects in other clusters can be included.

$$\begin{cases} \theta_o > \theta_n \\ \theta_n = \frac{1}{k}\sum_{i=1}^{3} k_i^n S_i \\ \theta_o = \frac{1}{k}\sum_{i=1}^{3} k_i^o S_i \\ k = \sum_{i=1}^{3} k_i^o = \sum_{i=1}^{3} k_i^n \end{cases} \tag{5}$$

$$k^{upper} = min\{min\{k|B \in \psi_n^k\}, min\{k|A \in \psi_o^k\}\} \tag{6}$$

$$k^{lower} = min\{k|k_2^o > k_2^n + \frac{S_4 - S_1}{S_2 - S_3}\} \tag{7}$$

D. Discussion

In this section, we discuss the proposed NA and conceptually related techniques to show that the proposed NA is novel and fundamentally different. One is the k-NN classifier, which also looks for neighborhood objects when classifying objects. The difference is that the k-NN classifier is a supervised method, but NA is not.

Another related technique is the mean-shift technique [7], which is also widely applied in image processing [35]. NA can be repeated several times and the process iteratively replaces an object's outlier score with its neighbors' mean scores. This process is close to the mean-shift process [11]. The difference is that mean-shift modifies the feature values of the objects whereas NA modifies the outlier score values of the objects.

All k-NN-based outlier detectors are related as they use k-NN as their key component. However, their usage of k-NN differs. In general, all k-NN-based detectors use k-NN to *produce* the outlier scores for the objects, as shown at the top of Figure 10. However, NA uses k-NN to *revise* the outlier scores produced by any detector, including detectors based on k-NN, as shown at the bottom of Figure 10. Using k-NN as a *detector* to produce outlier scores is a well-known approach but it is novel to use it as a *post-processing technique* for tuning the score.

Outlier scores of detector d:	$\mathrm{S^d} = f^{\mathrm{d}}(\mathrm{data})$
Revised outlier Score of ensemble:	$\mathrm{R} = g^{\mathrm{d}}(\mathrm{S}^1, \ldots, \mathrm{S}^{\mathrm{d}})$
Revised outlier Score of NA:	$\mathrm{R^d} = h^{\mathrm{d}}(\mathrm{data}, \mathrm{S^d})$

Fig. 10. Difference between k-NN-based detectors, ensembles, and NA: they require different inputs and have different models.

It is worth noting that other detectors [7], [14], [16] also utilize k-NN and the *average* operation. However, these are stand-alone detectors and cannot be an add-on to existing detectors, while NA is an add-on to other existing detectors and cannot be used as a stand-alone detector.

Ensemble techniques are also related and have a *combination* operation. Besides this commonality, NA has three fundamental differences. First, ensemble techniques combine several poor detectors to obtain a better one [1], as shown in the revised outlier score in the ensemble in Figure 10, while NA removes local variance. Second, ensemble techniques need to compute the outlier score for the same object multiple times, while NA does not. Third, ensemble techniques cannot be applied to a single detector, but NA can.

NA and ensemble techniques are not exclusive, and they can be applied jointly. Their similarity is that both aim to smooth the outlier scores; the ensemble operates across the detectors while NA operates across the objects. Considering the two detectors (the blue and green lines) in Figure 5, ensemble techniques can improve these two poorly performing detectors only when the two peaks happen in the same location (objects) and with a proper difference.

It is worthwhile to note that NA may be suitable for other score-based data-mining tasks. This is because similar input should have a similar output. If we extend the definition of ensemble as the technique having the operation of score *combination*, we can identify several types of ensembles. These types include feature-based ensemble (feature bagging), detector-based ensemble, parameter-based ensemble, and object-based ensemble (the proposed NA). These ensembles should be applicable to data-mining tasks other than outlier detection.

Recently, Ke *et al*. [36] proposed a method called group similarity system (GSS) for unsupervised outlier detection and Yang *et al*. [37] proposed a data pre-processing technique called neighborhood representative (NR) to detect collective outliers using exiting outlier detectors. GSS partitions the data into non-overlapped groups and judges the groups as outliers by considering the mean of the outlier scores of the objects in each group. NR scores the representative objects sampled from each group and judges the groups as outliers by considering the scores of the representative objects in each group. NA

is not used for collective outliers but for individual outliers, making it different from GSS or NR.

IV. EXPERIMENTAL SETUP

We used nine public, real-world, semantically meaningful static datasets, which can be found in UCI repository datasets or [22]. The information in the datasets varies from 8 to 259. They contain outliers ranging from 0.40% to 75.40% and have objects ranging from 195 to 60,632 as summarized in Table I. For preprocessing, all data were scaled by subtracting the mean and dividing by the standard deviation for each attribute.

TABLE I
DATASET INFORMATION

Name	Objects	Dimension	Outlier	Outlier objects	Normality objects
KDD-Cup99	60632	38	246	Intrusion connections	Normal connections
Stamps	340	9	340	Forged stamps	Genuine stamps
PageBlocks	5393	10	510	Text block	Other types of block
Cardiotocography	2114	21	466	Suspect or pathological people	Healthy people
Pima	768	8	268	People with diabetes	Healthy people
SpamBase	4207	57	1679	Spam emails	Non-spam emails
HeartDisease	270	13	120	People with heart problem	Healthy people
Arrhythmia	450	259	206	Patients with arrhythmia	Healthy people
Parkinson	195	22	147	Patients with Parkinson's disease	Healthy people

The outlier detectors' performance was measured mainly by the area under the receiver operating characteristic (ROC) curve (AUC). The ROC curve was drawn by plotting the true positive rate against the false positive rate over various threshold values. The AUC was a single value ranging from 0 to 1. The bigger the value was, the better the performance.

While AUC measured the average performance, we also tested the performance when a selected thresholding method was applied. For the threshold component, we used the known number of outliers in the dataset. This is known as the top-k method. The result was measured by the F1-score, which was the average of precision and recall. Precision is the ability to minimize false positives and recall is the ability to find all the positive samples.

For k-NN-based outlier detectors, we used the value of k, which provided the best results when k ranged from 2 to 100. Further discussion on the identification of the optimal parameter k value for k-NN-based detectors can be found in this comprehensive survey [38]. The default parameters found in the literature are used for the other detectors.

The proposed NA was tested with all values of k from 1 to 100. We used k = 100 as the default value. NA was iterated 10 times to study the effect of $iterations$.

V. RESULTS

A. The overall effect of NA

We varied the neighborhood size k in NA from 1 to 100 to find the best results and compared them with the results obtained using the default value k = 100. The average AUC and F1-score results are summarized in Table II. The AUC results per dataset are summarized in Table III. Based on the results, we can make the following observations.

First, based on the AUC results in Table II, the proposed NA significantly improved all the detection results. On average, all the detectors evaluated for all the datasets improved by +0.04 (from 0.70 to 0.74) with the default k, and +0.06 with the best k. We can make a similar observation about AUC for the F1-score. NA improved all outlier detectors by +0.02 (from 0.73 to 0.75) on average when using the default value of k, and by +0.03 when using the best value of k.

Second, NA provided the most AUC improvement with the NC detector, from 0.62 to 0.77. The most significant individual improvement was +0.28 for *HeartDisease* and *KDD-Cup99*. This observation is interesting, as NC was originally one of the worst detectors. However, when used with NA, it became competitive. This indicates that NC and NA utilize different properties and are complementary. It also suggests that the poorly performing detectors evaluated previously may have been seriously underestimated.

Third, the default setting with k = 100 performed almost as well as the best k. This shows that NA is robust on the choice of the parameter k.

Fourth, as shown in the columns of data labeled original in Table II, except MO_GAAL, without using NA the average AUC of detectors has a range from 0.62 to 0.75. However, with NA, the range becomes much smaller, from 0.75 to 0.79. This indicates that when NA was not used, the choice of detector mattered, but when NA was used it mattered less. This may pose a challenge to the generally accepted idea over the decades that the data model is the most important factor [1]. For MO_GAAL, the ROC AUC is near 0.50, which is close to random guesses. This may be because MO_GAAL needed more samples to train the neural network.

In Table III, we can see that all detectors for all datasets improved for both the default k and the best k. The only exception is the result for *Arrhythmia*, which weakened by -0.02 when using default k. Most datasets improved from +0.03 to +0.15 on average. The most significant individual improvement was for *HeartDisease*, which was +0.17 on average. NA did not help much with the datasets containing only a few outliers or when the original detector already performed well. For example, MOD, KNN, IFOREST, OCVSCM, and PCAD all achieved AUC = 0.99 for *KDD-Cup99*.

B. NA's effect on the best detector per dataset

The table presented in Table IV provides a summary of the best detectors with and without NA for each dataset. The impact of NA on the performance of the best detector varies across three distinct groups. The first group consists of datasets, such as *Cardio.* and *SpamBase*, where the best detectors remain the same when NA is used. The second group includes datasets like *PageBlocks*, *Pima*, *HeartDisease*, and *Parkinson*, where the best detectors completely change when NA is applied. The third group comprises datasets such as *KDD-Cup99*, *Stamps*, and *Arrhythmia*, where the best detectors partially change when NA is utilized. These findings suggest that there is no single detector that consistently outperforms others, regardless of the presence of NA. This observation aligns with the conclusion drawn by Aggarwal [1].

C. Effect of the iterations

NA can be iterated several times. Next, we varied the $iteration$ parameter from 1 to 10 times to study its effect on

TABLE III
AUC IMPROVEMENT PER OUTLIER DETECTOR PER DATASET

	Dataset	KDD-Cup99	Stamps	PageBlocks	Cardio.	Pima	SpamBase	HeartDisease	Arrhythmia	Parkinson	AVG.	DIFF.
	outliers	0.4%	9.1%	10.2%	22.2%	34.9%	39.4%	44.4%	45.8%	75.4%		
Original	MOD	0.99	0.90	0.91	0.54	0.68	0.55	0.62	0.74	0.64	0.73	-
	LOF	0.84	0.89	0.91	0.59	0.69	0.49	0.67	0.73	0.60	0.71	-
	ODIN	0.81	0.83	0.79	0.61	0.63	0.52	0.61	0.72	0.53	0.67	-
	NC	0.69	0.68	0.70	0.57	0.57	0.55	0.58	0.67	0.56	0.62	-
	KNN	0.99	0.91	0.92	0.55	0.73	0.57	0.68	0.74	0.66	0.75	-
	ABOD	0.86	0.87	0.85	0.48	0.70	0.42	0.65	0.73	0.64	0.69	-
	MCD	0.97	0.85	0.92	0.49	0.68	0.46	0.64	0.72	0.64	0.71	-
	IFOREST	0.99	0.86	0.90	0.70	0.67	0.64	0.65	0.76	0.47	0.74	-
	OCSVM	0.99	0.87	0.91	0.70	0.62	0.53	0.58	0.74	0.43	0.71	-
	PCAD	0.99	0.90	0.90	0.75	0.63	0.55	0.62	0.73	0.38	0.72	-
	MO_GAAL	0.55	0.63	0.56	0.50	0.56	0.73	0.41	0.50	0.50	0.55	-
	COPOD	0.99	0.93	0.88	0.66	0.65	0.69	0.69	0.76	0.54	0.75	-
	AVG	0.89	0.84	0.85	0.60	0.65	0.56	0.62	0.71	0.55	0.70	-
NA (default)	MOD	0.99	0.93	0.91	0.52	0.76	0.59	0.77	0.72	0.72	0.77	+0.04
	LOF	0.88	0.93	0.94	0.58	0.76	0.67	0.78	0.73	0.55	0.76	+0.05
	ODIN	0.83	0.93	0.83	0.74	0.74	0.57	0.75	0.70	0.56	0.74	+0.07
	NC	0.97	0.92	0.86	0.80	0.61	0.61	0.85	0.70	0.32	0.74	+0.12
	KNN	0.88	0.92	0.90	0.52	0.77	0.61	0.82	0.72	0.73	0.76	+0.01
	ABOD	0.85	0.90	0.79	0.41	0.78	0.36	0.85	0.70	0.80	0.72	+0.03
	MCD	0.99	0.92	0.93	0.47	0.74	0.43	0.81	0.70	0.76	0.75	+0.04
	IFOREST	0.98	0.92	0.87	0.74	0.73	0.65	0.81	0.70	0.54	0.77	+0.03
	OCSVM	0.98	0.92	0.90	0.75	0.68	0.56	0.72	0.72	0.56	0.75	+0.05
	PCAD	0.98	0.92	0.89	0.81	0.69	0.58	0.76	0.70	0.40	0.75	+0.03
	MO_GAAL	0.58	0.80	0.42	0.50	0.52	0.74	0.63	0.50	0.50	0.58	+0.03
	COPOD	0.95	0.93	0.84	0.68	0.70	0.69	0.82	0.71	0.60	0.77	+0.01
	AVG	0.90	0.91	0.84	0.63	0.71	0.59	0.78	0.69	0.59	0.74	+0.04
NA (Best)	MOD	0.99	0.95	0.92	0.55	0.76	0.59	0.77	0.74	0.74	0.78	+0.05
	LOF	0.88	0.94	0.94	0.59	0.76	0.68	0.78	0.74	0.60	0.77	+0.06
	ODIN	0.83	0.94	0.83	0.74	0.74	0.57	0.75	0.72	0.58	0.74	+0.07
	NC	0.97	0.92	0.86	0.80	0.61	0.62	0.86	0.72	0.58	0.77	+0.15
	KNN	0.99	0.95	0.92	0.55	0.77	0.61	0.82	0.74	0.75	0.79	+0.04
	ABOD	0.93	0.94	0.85	0.48	0.78	0.39	0.86	0.73	0.80	0.75	+0.06
	MCD	0.99	0.92	0.93	0.49	0.74	0.48	0.81	0.73	0.82	0.77	+0.06
	IFOREST	0.99	0.94	0.90	0.74	0.73	0.65	0.81	0.76	0.54	0.78	+0.05
	OCSVM	0.99	0.93	0.91	0.75	0.68	0.56	0.73	0.74	0.56	0.76	+0.05
	PCAD	0.99	0.94	0.90	0.81	0.69	0.58	0.77	0.73	0.40	0.76	+0.04
	MO_GAAL	0.60	0.80	0.56	0.50	0.57	0.74	0.63	0.50	0.50	0.60	+0.05
	COPOD	0.99	0.95	0.88	0.69	0.70	0.71	0.82	0.76	0.60	0.79	+0.03
	AVG	0.93	0.93	0.87	0.64	0.71	0.60	0.78	0.72	0.62	0.76	+0.06

TABLE II
AVERAGE AUC AND F1-SCORE FOR ALL DATASETS

Measurement	AUC			F1-score		
Detector	Original	NA (k)		Original	NA (k)	
		Default	Best		Default	Best
MOD [7]	0.73	0.77	0.78	0.74	0.76	0.77
LOF [20]	0.71	0.76	0.77	0.74	0.76	0.78
ODIN [16]	0.67	0.74	0.75	0.71	0.75	0.76
NC [18]	0.62	0.74	0.77	0.71	0.75	0.77
KNN [15]	0.75	0.76	0.79	0.74	0.76	0.77
ABOD [28]	0.69	0.72	0.75	0.73	0.75	0.75
MCD [23]	0.71	0.75	0.77	0.72	0.75	0.76
IFOREST [24]	0.74	0.77	0.79	0.74	0.76	0.77
OCSVM [26]	0.71	0.75	0.76	0.71	0.75	0.76
PCAD [27]	0.72	0.75	0.76	0.73	0.75	0.75
MO_GAAL [29]	0.55	0.58	0.60	0.67	0.69	0.69
COPOD [31]	0.75	0.77	0.79	0.76	0.77	0.78
AVG	0.70	0.74	0.76	0.73	0.75	0.76

TABLE IV
BEST DETECTOR WITH AND WITHOUT NA FOR EACH DATASET

Dataset	without	with NA
KDD-Cup99	MOD, KNN, IFOREST OCSVM, PCAD, COPOD	MOD, KNN, MCD, IFOREST OCSVM, PCAD, COPOD
Stamps	COPOD	MOD, KNN ,COPOD
PageBlocks	KNN, MCD	LOF
Cardio.	PCAD	PCAD
Pima	KNN	ABOD
SpamBase	MO_GAAL	MO_GAAL
HeartDisease	COPOD	NA, ABOD
Arrhythmia	COPOD	IFOREST, COPOD
Parkinson	KNN	MCD

the result. The value iteration = 0 corresponds to the original detector without NA. The average AUC results of all detectors evaluated for all datasets, a selected detector (MOD), and a selected dataset (*HeartDisease*) are summarized in Table V, Table VI, and Table VII, respectively.

The average results in Table V show the first iteration achieved the most improvement (+0.06). The second iteration achieved further improvement (+0.01) but beyond that, the effect remained rather small ($\leq$+0.03). However, by applying NA for multiple iterations the performance was improved from 0.70 to 0.79 AUC.

However, it has been observed that as the number of iterations increases, the scores assigned to all objects tend to converge. This convergence negatively impacts the performance of object detection, as indicated by performance drop (NC) and fluctuations (OCSVM). Achieving an optimal number of iterations is challenging in unsupervised learning. To mitigate this issue, it is not recommended to iterate NA (number of iterations) excessively. However, it is worth noting that a significant improvement in performance is observed when using a single iteration for all detectors, thus it is considered safe to use NA with a single iteration.

The results for the individual datasets with MOD are

TABLE V
AVERAGE AUC RESULTS FOR ALL DATASETS

	Iteration										
Detector	0	1	2	3	4	5	6	7	8	9	10
MOD	0.73	0.78	0.8	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81
LOF	0.71	0.77	0.78	0.79	0.80	0.80	0.80	0.80	0.80	0.80	0.80
ODIN	0.67	0.75	0.77	0.78	0.79	0.79	0.80	0.80	0.80	0.80	0.80
NC	0.62	0.77	0.76	0.76	0.75	0.75	0.74	0.74	0.74	0.74	0.74
KNN	0.75	0.79	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81
ABOD	0.69	0.75	0.78	0.79	0.79	0.79	0.79	0.79	0.79	0.79	0.79
MCD	0.71	0.77	0.78	0.79	0.79	0.79	0.79	0.79	0.79	0.79	0.79
IFOREST	0.74	0.79	0.80	0.82	0.82	0.82	0.82	0.82	0.82	0.82	0.83
OCSVM	0.71	0.76	0.79	0.81	0.82	0.82	0.81	0.82	0.82	0.82	0.82
PCAD	0.72	0.76	0.77	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78
MO_GAAL	0.55	0.60	0.61	0.61	0.61	0.61	0.61	0.61	0.61	0.61	0.62
COPOD	0.76	0.79	0.82	0.83	0.84	0.84	0.84	0.85	0.85	0.85	0.85
AVG	0.70	0.76	0.77	0.78	0.78	0.78	0.78	0.79	0.79	0.79	0.79

TABLE VI
AUC RESULTS OF MOD DETECTOR

	Iteration										
Dataset	0	1	2	3	4	5	6	7	8	9	10
KDD-Cup99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99
Stamps	0.90	0.95	0.95	0.95	0.96	0.96	0.96	0.95	0.95	0.95	0.95
PageBlocks	0.91	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.92
Cardio.	0.54	0.55	0.55	0.55	0.55	0.55	0.55	0.55	0.55	0.56	0.56
Pima	0.68	0.76	0.78	0.78	0.77	0.76	0.76	0.75	0.75	0.75	0.75
SpamBase	0.55	0.59	0.65	0.69	0.72	0.74	0.75	0.75	0.76	0.76	0.77
HeartDisease	0.62	0.77	0.86	0.89	0.90	0.90	0.90	0.91	0.91	0.91	0.91
Arrhythmia	0.74	0.74	0.71	0.69	0.67	0.65	0.65	0.64	0.64	0.64	0.64
Parkinson	0.64	0.74	0.80	0.83	0.84	0.84	0.85	0.85	0.85	0.85	0.85
AVG	0.73	0.78	0.80	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81

TABLE VII
AUC RESULTS ON *HEARTDISEASE* DATASET

	Iteration										
Detector	0	1	2	3	4	5	6	7	8	9	10
MOD	0.62	0.77	0.86	0.89	0.9	0.9	0.9	0.91	0.91	0.91	0.91
LOF	0.67	0.78	0.85	0.89	0.90	0.90	0.90	0.91	0.91	0.91	0.91
ODIN	0.61	0.75	0.85	0.89	0.90	0.90	0.90	0.90	0.91	0.91	0.91
NC	0.58	0.86	0.90	0.91	0.91	0.91	0.91	0.91	0.91	0.91	0.91
KNN	0.68	0.82	0.87	0.89	0.90	0.90	0.91	0.91	0.91	0.91	0.91
ABOD	0.65	0.86	0.89	0.90	0.90	0.90	0.91	0.91	0.91	0.91	0.91
MCD	0.64	0.81	0.87	0.89	0.90	0.90	0.91	0.91	0.91	0.91	0.91
IFOREST	0.65	0.81	0.87	0.89	0.90	0.90	0.91	0.91	0.91	0.91	0.91
OCSVM	0.58	0.73	0.84	0.89	0.90	0.90	0.90	0.91	0.91	0.91	0.91
PCAD	0.62	0.77	0.86	0.89	0.90	0.90	0.90	0.91	0.91	0.91	0.91
MO_GAAL	0.41	0.63	0.64	0.63	0.62	0.62	0.61	0.61	0.61	0.61	0.61
COPOD	0.69	0.82	0.88	0.90	0.90	0.90	0.91	0.91	0.91	0.91	0.91
AVG	0.62	0.78	0.85	0.87	0.88	0.88	0.88	0.88	0.89	0.88	0.89

reported in Table VI. All the datasets evaluated with the MOD detector were improved except *Arrhythmia*, which started to deteriorate during the second iteration. This might have been caused by the so-called curse of dimensionality in high-dimensional data, as *Arrhythmia* has 259 dimensions, while all the other datasets had 60 or fewer. Most other datasets were improved even when they were iterated 10 times. Another exception was *Pima*, for which the result started to deteriorate after the fourth iteration. This indicated that the iteration parameter needed to be tuned according to the datasets if desiring an optimal value. To be conservative, we set the default value as iteration = 1 despite knowing that some datasets, such as *SpamBase* and *HeartDisease*, would benefit from more iterations.

The results for the individual detectors with *HeartDisease* are reported in Table VII. It shows all detectors can benefit from iteration = 2.

Drawing inspiration from the elbow method, which is a visual technique for determining the optimal K in K-means clustering, we can construct an iteration-mean-squared-error (iteration-MSE) curve by plotting the iteration of applying NA and the MSE between the outlier scores in the current and previous iterations. The elbow point in this curve can potentially serve as an indicator for identifying the optimal iteration. In Figure 11, we present the iteration-MSE curves for three detectors, namely LOF, NC, and OCSVM, when tested with the *Parkinson*, *HeartDiease*, and *Spambase* datasets. The elbow points in these curves accurately correspond to the optimal iterations, which typically fall within a range of three iterations. Even in the case of OCSVM on the *Spambase* dataset, where the performance of OCSVM deteriorates after five iterations, the elbow point is observed to be around four, effectively capturing the trend of performance change with increasing iterations. This demonstrates the utility of the elbow point in the iteration-MSE curve as an indicator for determining the optimal iteration.

To summarize, it can be determined that the ideal number of iterations for applying NA is contingent upon the specific dataset and detector employed. The elbow point observed in the iteration-MSE curve may serve as a useful indicator for identifying the optimal iteration. Nevertheless, we suggest a conservative approach of utilizing a single iteration, as it strikes a balance between performance and stability.

TABLE VIII
AVERAGE AUC RESULTS FOR ALL DATASETS WITH K IN NA EQUALING TO K IN K-NN BASED DETECTORS

Detector	Original		NA	
	Default k	Best k	Default k	Best k
MOD	0.71	0.73	0.75	0.77
LOF	0.7	0.71	0.74	0.75
ODIN	0.66	0.67	0.73	0.74
NC	0.60	0.62	0.66	0.73
KNN	0.72	0.75	0.75	0.77
ABOD	0.68	0.69	0.71	0.74
AVG	0.68	0.70	0.72	0.75

D. Effect of k

To study the effect of k in NA, we varied it from 1 to 100. The average AUC values across all the datasets are shown in Figure 12. The results on a selected individual dataset (*HeartDisease*) are also shown in Figure 13. The value k = 1 corresponds to the original detector without NA.

The results show that when increasing k, all detectors improved and reached their best performance with k = 100. We, therefore, recommend k = 100 as the default value.

NA is proposed as an independent component to improve any single outlier detector. We notice that all k-NN-based outlier detectors also need to select the value of k. We considered using the same k value both for the baseline detectors and for NA. We performed additional experiments with the k-NN-based detectors. We varied k from 3 to 100 to find the best AUC.

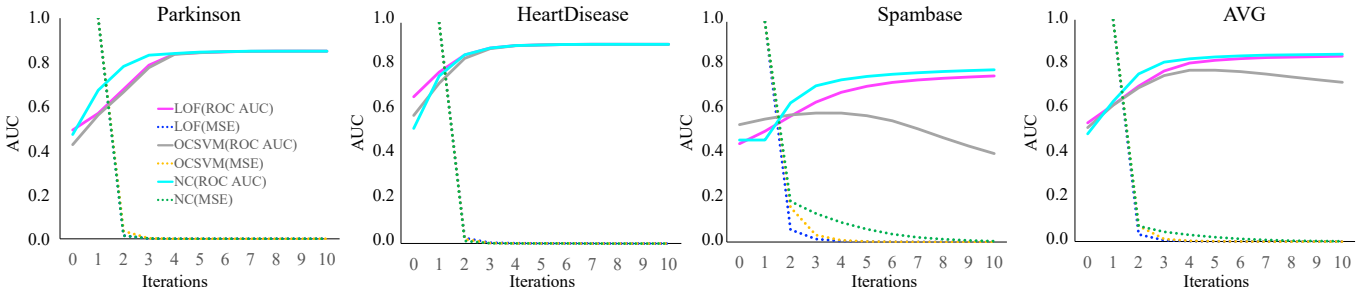


Fig. 11. The elbow points in the iteration-MSE curves for determining the optimal iteration of applying NA. The MSE at each iteration is normalized by dividing it by the maximum MSE value. The AVG represents the average results of the three tested datasets.

TABLE IX
AUC DIFFERENCE RESULTS FOR ALL DATASETS FOR AVERAGE JOINTLY WORKING WITH NA

Detector combination	Post-processing	KDD-Cup99	Stamps	PageBlocks	Cardio.	Pima	SpamBase	HeartDisease	Arrhythmia	Parkinson	AVG.
MOD+KNN	Average	0.99	0.91	0.92	0.55	0.73	0.57	0.68	0.74	0.66	0.75
	+NA (default k)	0.99	0.92	0.90	0.52	0.77	0.61	0.82	0.72	0.73	0.77
	+NA (best k)	0.99	0.95	0.92	0.55	0.77	0.61	0.82	0.74	0.75	0.79
ODIN+NC	Average	0.81	0.83	0.79	0.61	0.63	0.52	0.61	0.72	0.53	0.67
	+NA (default k)	0.81	0.93	0.83	0.74	0.74	0.57	0.75	0.70	0.56	0.74
	+NA (best k)	0.83	0.94	0.83	0.74	0.74	0.57	0.75	0.72	0.58	0.75
MOD+NC	Average	0.69	0.68	0.70	0.57	0.57	0.55	0.58	0.67	0.56	0.62
	+NA (default k)	0.87	0.92	0.86	0.80	0.61	0.61	0.85	0.70	0.32	0.73
	+NA (best k)	0.87	0.92	0.86	0.80	0.61	0.62	0.86	0.72	0.58	0.76
MOD+ODIN+NC+KNN	Average	0.99	0.90	0.91	0.54	0.68	0.55	0.62	0.74	0.64	0.73
	+NA (default k)	0.99	0.93	0.91	0.52	0.76	0.59	0.77	0.72	0.72	0.77
	+NA (best k)	0.99	0.95	0.92	0.55	0.76	0.59	0.77	0.74	0.74	0.78
All twelve detectors	Average	0.89	0.84	0.82	0.60	0.65	0.56	0.62	0.71	0.55	0.69
	+NA (default k)	0.91	0.91	0.82	0.63	0.71	0.59	0.77	0.69	0.59	0.73
	+NA (best k)	0.93	0.93	0.84	0.64	0.71	0.60	0.77	0.72	0.62	0.75

The average results over all datasets are summarized in Table VIII. They show that NA significantly improved the detectors by +0.05 on average. Most improvement is achieved with NC (+0.11). Further minimal improvements might be achieved with some datasets if k was increased further. However, some datasets do not have enough data to go much beyond 100, and the results would eventually start to degrade. The main result was that we can achieve good performance with rather small k values.

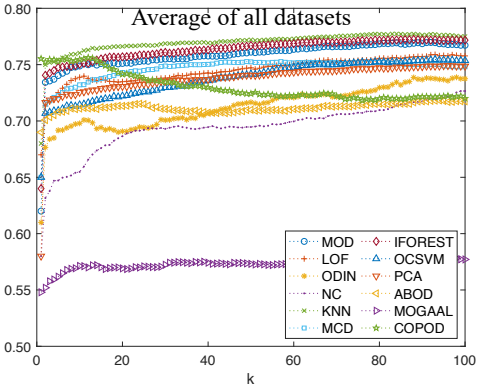


Fig. 12. Average AUC results for all datasets with varying k.

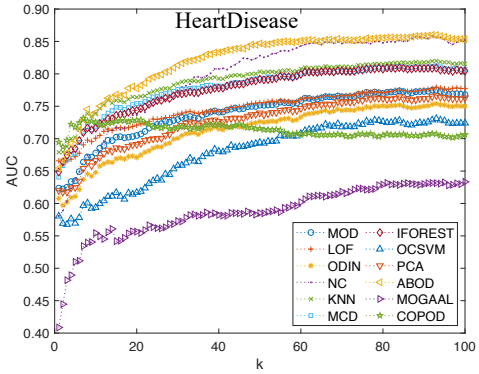
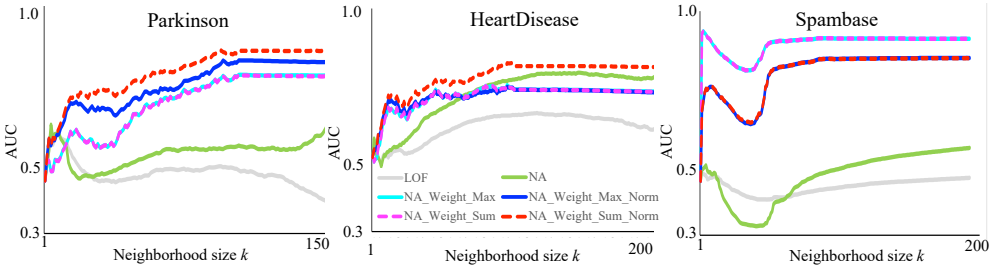


Fig. 13. AUC results on *HeartDisease* with varying k.

E. Effect of the neighbor weight

The effect of weight in Equation 1 is studied in this section. The W_j corresponding to $X_j \in \psi_i^k$ in Equation 1 can be calculated by considering the distance between X_j and X_i, denoted as $d_{i,j}$. Hence, W_j and the normalized W_j can be calculated as $W_j = W_j^*$ and $W_j = W_j^* / \sum_c W_c^*, X_c \in \psi_i^k$, respectively. Here, $W_j^* = 1/d_{i,j}$ when $d_{i,j} \neq 0$ and $W_j^* = 1/\gamma$ when $d_{i,j} = 0$. γ can be $\gamma = \sum_c d_{i,c}, X_c \in \psi_i^k$ (denoted as Sum) or $\gamma = max\{d_{i,c} | X_c \in \psi_i^k\}$ (denoted as Max). Hence, four ways of calculating W_j can be obtained by considering the combination of normalization and γ.

Figure 14 shows the results of LOF detector tested



Best AUC per dataset per method

	LOF	NA	NA Weight Max	NA Weight Max Norm	NA Weight Sum	NA Weight Sum Norm
Parkinson	0.60	0.64	0.78	0.83	0.78	0.86
HeartDisease	0.67	0.80	0.76	0.75	0.76	0.83
Spambase	0.49	0.57	0.94	0.85	0.94	0.85
AVG	0.59	0.67	0.83	0.81	0.83	0.85

Fig. 14. The impact of various methods weighting NA is studied using three different datasets. The average outcomes of these three datasets are denoted as AVG.

with the *Parkinson*, *HeartDiease*, and *Spambase* datasets. *NA_Weight_Max* and *NA_Weight_Max_Norm* are for $W_j = W_j^*$ and $W_j = W_j^* / \sum_c W_c^*, X_c \in \psi_i^k$ with $\gamma = max\{d_{i,c} | X_c \in \psi_i^k\}$, respectively. *NA_Weight_Sum* and *NA_Weight_Sum_Norm* are for $W_j = W_j^*$ and $W_j = W_j^* / \sum_c W_c^*, X_c \in \psi_i^k$ with $\gamma = \sum_c d_{i,c}, X_c \in \psi_i^k$, respectively. The choice of the optimal weighting method is contingent upon the specific dataset. It is observed that all weighting methods yield substantial improvements in the *Parkinson* and *Spambase* datasets compared to NA. However, in the case of the *HeartDiease* dataset, only the *NA_Weight_Sum_Norm* method demonstrates the ability to enhance NA. Consequently, we recommend this method as the default weighting approach. Further research on the various weighting methods could be considered as a potential avenue for future investigation.

F. Effect of the distance metric

The effect of defining similar objects using different distance metrics, which is the core concept of NA, is studied in this section. Figure 15 shows the results of LOF (k=100) and OCSVM detector tested with the *Parkinson*, *HeartDiease*, and *Spambase* datasets. In the legend of Figure 15, the letters E and V represent fixing the distance metric to the *Euclidean* metric and varying the distance metric among *Braycurtis*, *Canberra*, *Chebyshev*, *Euclidean*, and *Manhattan*, respectively.

Several observations can be deduced from the results. Firstly, the $LOF(V)\_NA(V)$ consistently exhibits superior performance across various datasets. This suggests that the selection of the distance metric can align with the requirements of the detector, which relies on a distance metric to generate outlier scores. Secondly, for the *HeartDiease* dataset, the choice of distance metric does not significantly impact the results. However, for the *Parkinson* and *Spambase* datasets, the distance metric has a substantial influence on the outcomes. This indicates that the selection of the distance metric is contingent upon the specific application. Thirdly, the utilization of the *Braycurtis*, *Euclidean*, and *Manhattan* distance metrics can enhance the performance of both the LOF and OCSVM detectors across all datasets. Based on the average results, namely AVG, the LOF detector benefits the most from the adoption of the *Manhattan* distance metric, while the OCSVM detector experiences the greatest improvement with the use of the *Braycurtis* distance metric. Notably, the *Euclidean* distance metric brings about equal enhancements for both detectors. This suggests that the choice of distance metric can be determined by the requirements of the specific detector. In other words, the *Manhattan* and *Braycurtis* metrics can be employed as practical solutions for neighborhood-based and non-neighborhood-based detectors, respectively. On the other hand, the *Euclidean* metric can be utilized for combining both neighborhood and non-neighborhood-based detectors in an ensemble. Additional research on the distance metric could be explored as a potential avenue for future inquiry.

G. Outlier ensembles

Next, we tested the effect of augmentation on NA with an existing outlier ensemble technique. We used the average ensemble [1] method, with different baseline detector combinations. Results are summarized in Table IX.

We can observe that the results of the outlier ensemble depend on the quality of the individual detectors. The best results are obtained by the combination of MOD and KNN, which reaches 0.75. Combining all 12 detectors would reach only 0.69.

When applying NA jointly with the outlier ensemble, we observed the following. First, no matter which combination was used, NA always improved the results of the ensemble. Second, the best combination no longer depended on the quality of the individual detector. The best combination (MOD and KNN) is based on one of the weaker baseline detectors among those tested. This combination with NA reached the overall best result of 0.79, which was very close to the result (0.77) reached without optimizing the parameter k. This indicates that NA provides a strong complementary component to ensembles.

H. Complementary to NR

As in our previous work, NR was a data preprocessing method to improve detectors, we wanted to know if NA as an outlier score post-processing method could further improve NR. We tested LOF, NR+LOF, LOF+NA, and NR+LOF+NA by setting their parameter k to be the same value. The results with *Parkinson*, *HeartDiease*, and *Spambase* datasets were plotted in Figure 16. From Figure 16, we could observe that a relatively larger k was good when NR and NA were jointly used. NR+NA+LOF could further improve NR+LOF 31% on average (0.88 vs. 0.71 AUC) relatively as shown on the right of Figure 16. It was noteworthy that the performance of LOF

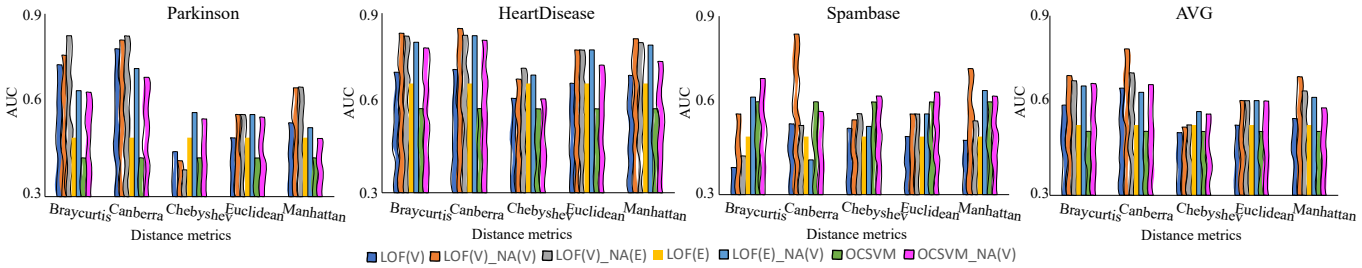
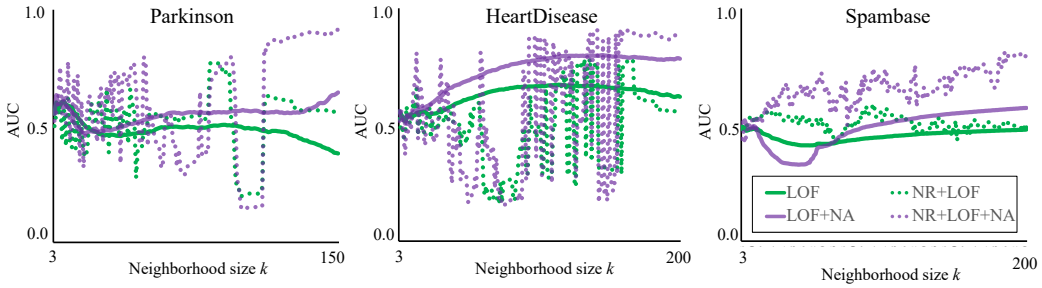


Fig. 15. The impact of various distance metrics on NA is examined using three distinct datasets. The average outcomes of these three datasets are denoted as *AVG*. In the legend, the symbols E and V denote the fixed utilization of the Euclidean metric and the variable utilization of the distance metric, encompassing the *Braycurtis*, *Canberra*, *Chebyshev*, *Euclidean*, and *Manhattan* metrics.



Best AUC per dataset per method

	LOF	NR+ LOF	LOF+ NA	NR+ LOF+ NA
Parkinson	0.60	0.77	0.64	0.92
HeartDisease	0.67	0.78	0.80	0.91
Spambase	0.49	0.59	0.57	0.81
AVG	0.59	0.71	0.67	0.88

Fig. 16. Experiment results of LOF, NR+LOF, LOF+NA, and NR+LOF+NA ranging k. NA is complementary to NR.

on *Spambase* dataset with 0.49 AUC was close to random guess, but when jointly used with NR and NA, it could even achieve 0.81 AUC. Another noteworthy finding was that the NR+LOF+NA approach exhibited a performance exceeding 0.90 AUC for both the *Parkinson* and *HeartDiease* datasets, in contrast to the LOF method which yielded AUC values below 0.70. This performance was significantly superior to any previously reported results of unsupervised outlier detectors in existing literature, as far as our knowledge extends. To summarize, NA is complementary to NR significantly.

I. Computational complexity

NA requires O(NlogN) calculations using KD-tree in low dimensions ($D < 20$) and Ball-tree in higher dimensions ($D > 20$) to find k-NN. However, since NA serves as a post-processing step, we care more about its gain relative to its additional cost. Table X shows the average extra computing time and the average AUC improvement over all datasets.

Table X shows that the k-NN-based detectors need only 4% extra time but can improve by 11% in AUC on average. Non-k-NN-based detectors are usually significantly faster and need 2,543% extra time to reach an average improvement of 7% in AUC. The main reason is that the k-NN-based detectors have already calculated the k-NN, which NA can directly utilize.

J. Discussion and limitations

Discussion: NA is not meant to be a stand-alone detector; rather, it is an add-on to any existing score-based outlier detector used to enhance its performance as shown in the example in Figure 3. The add-on does not increase the complexity of k-NN-based detectors as shown in section V.E, but it can bring significant improvement as shown in section V.A. NA has only one parameter k to tune, which is not sensitive (not oscillating) to detectors or datasets, and it is easy to tune as demonstrated in Section V.C. Hence, NA is very useful for practical applications.

TABLE X
AVERAGE AUC IMPROVEMENT AND EXTRA COMPUTING TIME USING NA FOR ALL DATASETS

Detectors		AUC	Extra
Category	Name	improvement(%)	time(%)
K-NN based	MOD	7	2
	LOF	8	5
	ODIN	11	4
	NC	24	2
	KNN	5	5
	ABOD	11	5
	AVG	11	4
Other	MCD	8	408
	IFOREST	6	891
	OCSVM	8	32
	PCAD	6	98
	MO_GAAL	6	> 1
	COPOD	5	1583
	AVG	7	2543

Limitations: One limitation of the method is the k-NN graph. Some neighbors can be far away, and simple averaging may not be the best solution. Possible alternatives could be to use the medoid or the weighted average. Different neighbor graphs [39]–[41] could also be used. Nevertheless, NA is already successful and we leave these ideas for future work.

NA also has the same limitation as other distance-based methods: its performance starts to degrade when the dimensions are large, as shown in the 269-dimensional *Arrhythmia* dataset. NA still improved but the performance started to

degrade if NA was iterated more than once. Such problems are common for distance-based pattern recognition methods operating in the raw attribute space. This is often referred to as the *curse of dimensionality*.

VI. CONCLUSIONS

A novel post-processing technique called neighborhood averaging (NA) for neighborhood smoothing in outlier score space is proposed. The technique can be used to improve any existing single outlier detector by smoothing its outlier scores. Simulations showed that it significantly improved all 12 tested outlier detectors including deep-learning-based detectors from 0.70 to 0.79 AUC on average. This has evidenced the importance of neighborhood smoothing in outlier score space.

The technique does not require any complicated parameter tuning and k is the only parameter when applying NA with a single iteration. When used with a k-NN-based baseline detector, we do not need to recalculate the k-NN and use the existing one with the same k value as the detector. With non-k-NN-based detectors, setting the value of $k = 100$ was shown to provide good results for almost all datasets. It is worth noting that once NA is applied, even a poorly performing outlier detector becomes competitive. This can help practitioners as they have one less design component to consider.

Outlier detection is an important topic in data mining. In addition to its ability to detect outliers in static data, it can also handle dynamic cases such as time series. Therefore, it is useful for applications like audio and video content analysis. In general, whenever similarity between objects can be properly predefined, whether static or dynamic, the concept of the neighborhood can be applied. Therefore, the proposed NA can be applied to enhance performance consistently and significantly. NA has the potential to be widely adopted in a variety of applications in data mining and beyond.

ACKNOWLEDGMENTS

The authors would like to express their utmost appreciation to the reviewers and editors for their valuable comments in improving the quality of the paper.

REFERENCES

[1] Charu C. Aggarwal. *Outlier Analysis Second Edition*. Springer International Publishing, 2016.

[2] Jiawei Yang, Sylwan Rahardja, and Susanto Rahardja. Click fraud detection: Hk-index for feature extraction from variable-length time series of user behavior. In *2022 IEEE 32nd International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE, 2022.

[3] R. Nawaratne, D. Alahakoon, D. De Silva, and X. Yu. Spatiotemporal anomaly detection using deep learning for real-time video surveillance. *IEEE Transactions on Industrial Informatics*, 16(1):393–402, 2020.

[4] Jiawei Yang, Gulraiz Iqbal Choudhary, Susanto Rahardja, and Pasi Franti. Classification of interbeat interval time-series using attention entropy. *IEEE Transactions on Affective Computing*, 14(1):321–330, 2023.

[5] Jiawei Yang, Radu Mariescu-Istodor, and Pasi Fränti. Three rapid methods for averaging gps segments. *Applied Sciences*, 9(22):4899, 2019.

[6] Jiawei Yang, Xu Tan, and Sylwan Rahardja. Mipo: how to detect trajectory outliers with tabular outlier detectors. *Remote sensing*, 2022.

[7] Jiawei Yang, Susanto Rahardja, and Pasi Fränti. Mean-shift outlier detection and filtering. *Pattern Recognition*, 115:107874, 2021.

[8] Pasi Fränti and Jiawei Yang. Medoid-shift for noise removal to improve clustering. In *International Conference on Artificial Intelligence and Soft Computing*, pages 604–614. Springer, 2018.

[9] T. Zhang, H. Qiu, G. Castellano, M. Rifai, C. S. Chen, and F. Pianese. System log parsing: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 35(8):8596–8614, 2023.

[10] Hongzuo Xu, Guansong Pang, Yijie Wang, and Yongjun Wang. Deep isolation forest for anomaly detection. *IEEE Transactions on Knowledge and Data Engineering*, 35(12):12591–12604, 2023.

[11] Jiawei Yang, Susanto Rahardja, and Pasi Fränti. Outlier detection: how to threshold outlier scores? In *Proceedings of the international conference on artificial intelligence, information processing and cloud computing*, pages 1–6, 2019.

[12] Jiawei Yang, Sylwan Rahardja, and Susanto Rahardja. Foor: Be careful for outlier-score outliers when using unsupervised outlier ensembles. *IEEE Transactions on Computational Social Systems*, 2023.

[13] Jiawei Yang, Sylwan Rahardja, and Susanto Rahardja. Regional ensemble for improving unsupervised outlier detectors. *SSRN*, 2023.

[14] S. Shekhar, C. Lu, and P. Zhang. A unified approach to detecting spatial outliers. *GeoInformatica*, 7(2):139–166, 2023.

[15] Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim. Efficient algorithms for mining outliers from large data sets. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 427–438, 2000.

[16] Ville Hautamaki, Ismo Karkkainen, and Pasi Franti. Outlier detection using k-nearest neighbour graph. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 3, pages 430–433. IEEE, 2004.

[17] Edwin M. Knorr and Raymond T. Ng. Algorithms for mining distance-based outliers in large datasets. In *Proceedings of 24th international conference on very large databases (VLDB'98)*, pages 392–403, 1998.

[18] Xiaojie Li, Jiancheng Lv, and Zhang Yi. An efficient representation-based method for boundary point and outlier detection. *IEEE transactions on neural networks and learning systems*, 29(1):51–62, 2016.

[19] J.W. Yang, S. Rahardja, and P. Fränti. Mean-shift outlier detection. In *Int. Conf. Fuzzy Systems and Data Mining*, pages 208–215, 2018.

[20] Markus Breunig, Hans-Peter Kriegel, Raymond Ng, and Joerg Sander. Lof: Identifying density-based local outliers. *ACM Sigmod Record*, 29:93–104, 06 2000.

[21] W. Hu, J. Gao, B. Li, O. Wu, J. Du, and S. Maybank. Anomaly detection using local kernel density estimation and context-based regression. *IEEE Transactions on Knowledge and Data Engineering*, 32(2):218–233, 2020.

[22] G.O. Campos, A. Zimek, J. Sander, R.J.G.B. Campello, B. Micenkova, E. Schubert, I. Assent, and M.E. Houle. On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study. *Data Mining and Knowledge Discovery*, 30(4):891–927, 2016.

[23] Peter J Rousseeuw. Least median of squares regression. *Journal of the American statistical association*, 79(388):871–880, 1984.

[24] F. Liu, T. Ting, K. Ming, and ZH. Zhou. Isolation-based anomaly detection. *ACM Transactions on Knowledge Discovery from Data*, 6(1):3:1–3:39, 2012.

[25] Xu Tan, Jiawei Yang, and Susanto Rahardja. Sparse random projection isolation forest for outlier detection. *Pattern Recognition Letters*, 163:65–73, 2022.

[26] Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola, and Robert C Williamson. Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471, 2001.

[27] M-L. Shyu, S-C. Chen, K. Sarinnapakorn, and LW. Chang. A novel anomaly detection scheme based on principal component classifier. In *ICDM Foundation and New Direction of Data Mining workshop*, pages 172–179, 2003.

[28] H. Kriegel, M. Schubert, and A. Zimek. Angle-based outlier detection in high-dimensional data. In *the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 444–452, 2008.

[29] Yezheng Liu, Zhe Li, Chong Zhou, Yuanchun Jiang, Jianshan Sun, Meng Wang, and Xiangnan He. Generative adversarial active learning for unsupervised outlier detection. *IEEE Transactions on Knowledge and Data Engineering*, 32(8):1517–1528, 2019.

[30] Zheng Li, Yue Zhao, Xiyang Hu, Nicola Botta, Cezar Ionescu, and George Chen. Copod: Copula-based outlier detection. *IEEE International Conference on Data Mining*, pages 1118–1123, 2020.

[31] Zheng Li, Yue Zhao, Xiyang Hu, Nicola Botta, Cezar Ionescu, and George Chen. Ecod: Unsupervised outlier detection using empirical cumulative distribution functions. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[32] Charu C. Aggarwal. *Recommender Systems*. Springer, 2016.
[33] P. Fränti and S. Sieranoja. K-means properties on six clustering benchmark datasets. *Applied Intelligence*, 48(12):4743–4759, 2018.
[34] Charu C. Aggarwal and Saket Sathe. Theoretical foundations and algorithms for outlier ensembles. *SIGKDD Explor. Newsl*, 17(1):24–47, 2015.
[35] D. Comaniciu and P. Meer. Mean shift: a robust approach toward feature space analysis. *IIEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):1603–619, 2022.
[36] Wenjun Ke, Jianguo Wei, Naixue Xiong, and Qingzhi Hou. Gss: A group similarity system based on unsupervised outlier detection for big data computing. *Information Sciences*, 620:1–15, 2023.
[37] Jiawei Yang, Yu Chen, and Sylwan Rahardja. Neighborhood representative for improving outlier detectors. *Information Sciences*, 625:192–205, 2023.
[38] Jiawei Yang, Xu Tan, and Sylwan Rahardja. Outlier detection: How to select k for k-nearest-neighbors-based outlier detectors. *Pattern Recognition Letters*, 174:112–117, 2023.
[39] K.C. Gowda and G. G. Krishna. The condensed nearest neighbor rule using the concept of mutual nearest neighborhood. *IEEE Transactions on Information Theory*, 25(4):488–490, 1979.
[40] C. Zhong, D. Miao, and R. Wang. A graph-theoretical clustering method based on two rounds of minimum spanning trees. *Pattern Recognition*, 43(3):752–766, 2010.
[41] P. Fränti, R. Mariescu-Istodor, and C. Zhong. Xnn graph. pages 207–217. Springer, 2016.

Jiawei Yang (born in Shidian County) received a B.Eng degree in electronic engineering from Beihang University, China, in 2013, and M.Sc and Ph.D. degrees in computer science from the University of Eastern Finland in 2019 and 2020, respectively. *Closer to faith than to living with dignity* and *Places worth going are full of darkness and there are no shortcuts* are his life quotes.

Susanto Rahardja received his B.Eng. degree from the National University of Singapore, M.Eng. and Ph.D. degrees from Nanyang Technological University, Singapore; all in the field of Electrical and Electronic Engineering. He is a Ph.D. Advisor at the Northwestern Polytechnical University (NPU) and is a Professor at the Singapore Institute of Technology. His research interests are in multimedia coding and processing, wireless communications, discrete transforms, machine learning, signal processing algorithms, implementation and optimization. He contributed to the development of a series of audio compression technologies such as Audio Video Standards AVS-L, AVS-2, ISO/IEC 14496-3:2005/Amd.2:2006, and ISO/IEC 14496-3:2005/Amd.3:2006 which have been licensed worldwide. Dr. Rahardja has more than 15 years of experience in leading a research team in the above-mentioned areas. He was past Associate Editors of IEEE Transactions on Audio, Speech and Language Processing and IEEE Transactions on Multimedia, past Senior Editor of the IEEE Journal of Selected Topics in Signal Processing, and is currently serving as Associate Editors for the Elsevier Journal of Visual Communication and Image Representation IEEE Transactions on Industrial Electronics and a member of Editorial Board of IEEE Access. He was the Conference Chair of 5th ACM SIGGRAPHASIA in 2012, APSIPA Summit and Conferences in 2010 and 2018 as well as General Chair/co-chair in many other conferences/workshops in ACM, IEEE and SPIE. Dr Rahardja is a recipient of several honors including the IEE Hartree Premium Award, the Tan Kah Kee Young Inventors' Open Category Gold award, the Singapore National Technology Award, A*STAR Most Inspiring Mentor Award, Finalist of the 2010 World Technology and Summit Award, the Nokia Foundation Visiting Professor Award, the ACM Recognition of Service Award, etc. He is a Fellow of IEEE and a Fellow of the Academy Engineering, Singapore.

Pasi Fränti received his Master and Ph.D. degrees in Computer Science from the University of Turku, Finland in 1991 and 1994 respectively. He has been a tenure Professor at the University of Eastern Finland since 2000 where he is currently the leader of the machine learning group. He has published over 120 journals and 180 conference papers. His current research interests include clustering algorithms, location-based services, machine learning, web and text mining, and optimization of healthcare services. He is editor-in-chief of the AIMS Press journal Applied Computing and Intelligence.