Mugla Journal of Science and Technology

# THE ADJUSTED HISTOGRAM-BASED OUTLIER SCORE - AHBOS

Uğur BİNZAT, Department of Statistics, Dokuz Eylül University, Türkiye, ugur.binzat@deu.edu.tr
(https://orcid.org/0000-0003-1387-6661)
Engin YILDIZTEPE*, Department of Statistics, Dokuz Eylül University, Türkiye, engin.yildiztepe@deu.edu.tr
(https://orcid.org/0000-0002-7617-4934)

Received: 21.02.2023, Accepted: 25.06.2023 Research Article
*Corresponding author
DOI: 10.22531/muglajsci.1252876

**Abstract**

*Histogram is a commonly used tool for visualizing data distribution. It has also been used in semi-supervised and unsupervised anomaly detection tasks. The histogram-based outlier score is a fast unsupervised anomaly detection method that has become more popular because of the rapid increase in the amount of data collected in recent decades. Histogram-based outlier score can be computed using either static or dynamic bin-width histograms. When a histogram contains large gaps, the dynamic bin-width approach is preferred over the static bin-width approach. These gaps in a histogram usually occur as a result of various distributions in real data. When working with a static bin-width histogram, gaps can be utilized to acquire better distinction between outliers and inliers. In this study, we propose an adjusted version of the histogram-based outlier score named adjusted histogram-based outlier score, which considers neighboring bins prior to density estimation. Results from a simulation study and real data application indicate that the adjusted histogram-based outlier score yields a better performance not only in the simulated data but also for various types of real data.*
**Keywords: Unsupervised anomaly detection, Outlier, Histogram, Density estimation**

# DÜZELTİLMİŞ HİSTOGRAM TABANLI AYKIRI DEĞER PUANI

**Özet**

*Histogram verinin dağılımının gösteriminde yaygın olarak kullanılan bir yöntemdir. Ayrıca, yarı denetimli ve denetimsiz anomali tespiti için de kullanılmaktadır. Histogram tabanlı aykırı değer puanı, son yıllarda toplanan veri miktarındaki hızlı artış nedeniyle daha popüler hale gelen hızlı ve denetimsiz bir anomali belirleme yöntemidir. Histogram tabanlı aykırı değer puanı, statik veya dinamik kutu genişliğine sahip histogram kullanılarak hesaplanabilir. Histogramda kutular arası büyük boşluklar olduğunda, statik kutu genişliği yaklaşımı yerine dinamik kutu genişliği yaklaşımı daha iyi sonuçlar vermektedir. Histogramdaki kutular arası boşluklar gerçek verilerin geldiği çeşitli dağılımların bir sonucu olarak ortaya çıkabilmektedir. Statik histogram ile çalışırken, aykırı değerler ve olağan değerler arasında daha iyi bir ayrım elde etmek için bu boşluklardan yararlanılabilir. Bu çalışmada, yoğunluk kestirimi öncesinde komşu kutuları da dikkate alan düzeltilmiş histogram tabanlı aykırı değer puanı yöntemi önerilmiştir. Benzetim çalışması ve gerçek veri uygulamasından elde edilen sonuçlar, düzeltilmiş histogram tabanlı aykırı değer puanı yönteminin yalnızca yapay verilerde değil, aynı zamanda farklı türlerde gerçek verilerde de daha iyi bir performans sağladığını göstermektedir.*
**Anahtar Kelimeler: Denetimsiz anomali belirleme, Aykırı değer, Histogram, Yoğunluk kestirimi**

**Cite**

**Binzat, U., Yıldıztepe, E., (2023). "The Adjusted Histogram-Based Outlier Score - AHBOS", Mugla Journal of Science and Technology, 9(1), 92-100.**

## 1. Introduction

Anomaly detection has been the subject of extensive research in recent decades due to its broad range of applications. It is widely used in various fields, including intrusion detection, medical diagnosis, credit card fraud detection, fault detection in safety-critical systems, and enemy activity surveillance [1]. Since the onset of anomaly detection research, there have been many definitions of what constitutes an anomaly. Earlier definitions have posited an outlier as an irrelevant, deviated observation or generated by a different mechanism [2-4]. Breunig et. al [5] and Chandola et. al [1] have considered the definition of an anomaly within a broader spectrum by including neighborhood, pattern, and behavioral characteristics of data. In general, anomalies can be defined as data points that display different behavior from the majority of data. Anomalies may arise due to mechanical faults, changes in system behavior, fraudulent behavior, human error, instrument error, or just natural deviations in populations [6]. Anomaly detection approaches can be divided into three categories, which are supervised, semi-supervised, and unsupervised anomaly detection. Supervised anomaly detection methods require a fully pre-labeled training dataset tagged as normal or anomalous [1,6], and this approach is similar to imbalanced classification. However, most classifiers cannot incorporate strongly imbalanced data [7]. Nevertheless, labeling all data as normal or anomalous in real-life applications is almost impossible. Additionally, the semi-supervised approach requires a training dataset but only recognizes data tagged as normal. This scenario is also known as novelty detection or novelty recognition [6]. Semi-supervised approaches are suitable for the fraud detection application domain. Unsupervised anomaly detection assumes that anomalous instances are far more infrequent than normal in data. Unsupervised methods do not require pre-labeled training datasets, enabling them to be flexible for real-life applications [1,6,7]. For example, in intrusion detection, a study by Zoppi et al. [8] compared various anomaly detection algorithms to identify unknown or novel threats. The findings showed that unsupervised anomaly detection algorithms outperform other machine learning algorithms when threats are unknown. The result of an anomaly detection algorithm can be a label or a score; semi-supervised and unsupervised methods generally output a score that can be transformed into a label using a threshold [7].

Histogram and kernel-based approaches are the most common unsupervised anomaly detection methods, especially in network security and fraud detection domains [9-13]. In addition, the data to be processed in these domains is usually very large. The increasing amount of data and the expectation of a quick response in such applications (network security, sensor networks, fraud detection) increases the requirement for fast and accurate anomaly detection methods. Among well-known unsupervised anomaly detection methods, histogram-based approaches are both quick and competitive. While some state-of-the-art methods suffer from computation burden, histogram-based methods can complete tasks within minutes, even when applied to large-scale datasets [7,14]. Histogram-based methods can also be used in other application domains and anomaly detection scenarios. In 2012, Goldstein and Dengel [15] proposed a histogram-based scoring method called Histogram Based Outlier Score (HBOS). HBOS is an unsupervised multivariate anomaly detection method that computes an outlier score for each instance in data. According to various studies, HBOS provides competitive results for various anomaly detection scenarios [7,16,17]. Dobos et al. [18] compared various anomaly detection approaches, including HBOS, to detect non-random errors which are caused by measurement bias, instrument failures, or process leaks.

HBOS combines outlier scores obtained from the density of each feature of separate histograms. Histograms of these attributes are constructed using either a static bin-width or a dynamic bin-width approach, where the height of each bin represents a density estimate [15]. Data in anomaly detection tasks involves various distributions, and gaps can frequently occur in these datasets. Therefore, the static bin-width approach may produce poor density estimation when applied to datasets containing huge gaps between data instances. If the number of bins is incorrectly calculated, almost all data may be collected in a few bins. Goldstein and Dengel [15] have provided a dynamic bin-width approach applicable to this eventuality. Due to the mentioned reason, the dynamic bin-width approach is favored over the static bin-width approach. Even though the dynamic bin-width approach is designed to work well with these issues, it may suffer from pointwise data fluctuations. Generally, both approaches offer practical advantages depending on the data [15]. In Section 2, we demonstrated the impact of determining the appropriate number of bins in the presence of gaps. In such cases, employing a robust data-based rule for determining the number of bins offers an advantage over approaches depending on the sample size.

The static bin-width HBOS faces a potential limitation wherein inliers and outliers may receive equal density in a histogram. To overcome this issue, we propose an adjustment to the static bin-width HBOS involving the modification of densities by considering neighbor bins of the corresponding bin. The proposed method, Adjusted Histogram Based Outlier Score (AHBOS), efficiently utilizes gaps to detect outliers while successfully labeling inliers. According to real and simulated data outcomes, AHBOS generally yields better results.

Section 2 briefly covers existing methods in the related literature. The AHBOS is introduced in Section 3, followed by the numerical studies demonstrated in Section 4. Finally, Section 5 offers conclusions and discussions.

## 2. Methods

This section presents static and dynamic bin-width approaches and possible problems associated with the presence of anomalies in histograms. Subsequently, HBOS and the proposed approach AHBOS are provided.

### 2.1. Static and Dynamic Bin-Width Histograms

A static bin-width (traditional) histogram is one of the most common methods to demonstrate data distribution. The static bin-width histogram for a dataset with $N$ instances is constructed using $k$ equal-width bins. The number of $k$ should be pre-specified. The data instances falling into the bins determine the bin heights. These frequencies (heights) can also be considered to be density estimates.

Dynamic bin-width histograms differ in terms of having non-static bin-widths where $N/\mathrm{k}$ data instances are distributed equally into $k$ bins [15]. The area of each bin is equal since the number of samples that fall into the bins is equal. Every bin is rectangular, having a dynamic height and width. Data instances covering larger areas will eventually yield lower heights. In the dynamic bin-width approach, data instances are distributed into specified bins equally. However, as this is not applicable in some cases, the algorithm should allow more than $N/k$ instances in the bins. In addition to the area, the height of these bins should also grow appropriately. The remaining instances can be allocated to the bin where the

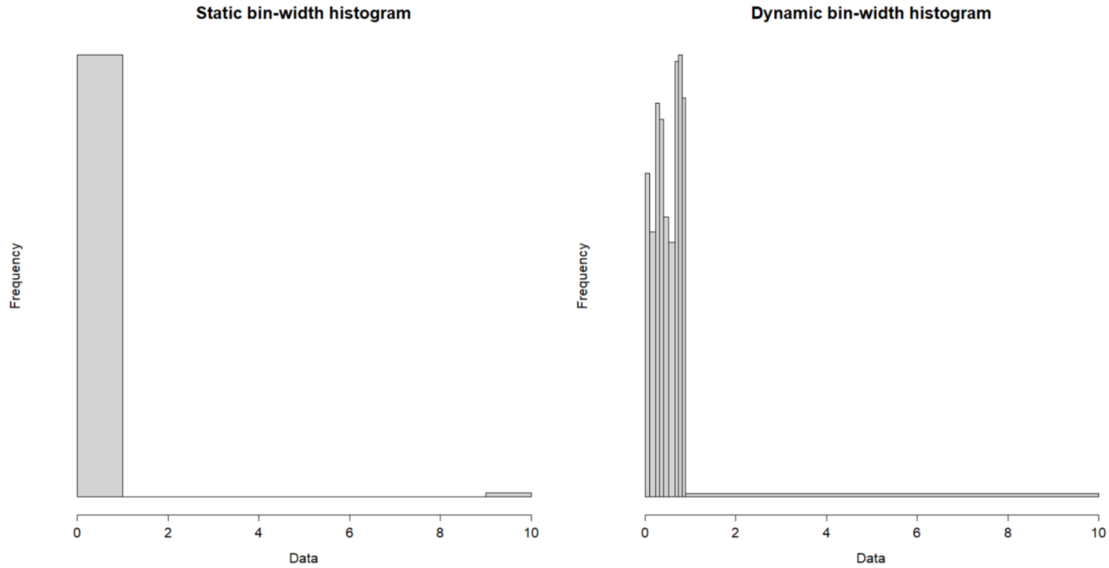


Figure 1. Static and dynamic bin-width histograms in the presence of an extreme value.

median is located. Likewise, the number of bins in the dynamic histogram cannot be larger than the unique values in the feature [19]. In this case, the number of bins diminishes to the number of unique values in the feature.

The critical point in constructing a histogram is correctly determining the bin number ($k$). Inappropriate $k$ can cause "under-smoothing" or "over-smoothing". Wand [20] discussed the choice of bin-width in detail and proposed a data-based choice. A common practice is to choose the square root of the $N$, where $N$ is the number of instances $x$ [15]. Sturges' rule [21] is the most well-known and oldest method to compute the number of bins. *hist()* function in the R statistical programming language [22] has three built-in options for setting the "break" which are "Sturges" [21] as default, "Scott" [23], and "FD" [24]. After computing bin-width with these rules, R uses a "pretty" function that ensures a good choice of bin number [20,25]. The aforementioned rules are provided in Table 1.

Table 1. Bin-width rules in the generic function *hist()* in R.

| Rule | Bin-width |
|---|---|
| Sturges (1926) | $\frac{\max(x) - \min(x)}{\lceil 1 + \log_2 N \rceil}$ |
| Scott (1979) | $\frac{3.49\hat{\sigma}}{\sqrt[3]{N}}$ |
| Freedman and Diaconis (1981) | $\frac{2\,IQR(x)}{\sqrt[3]{N}}$ |

Note: $\hat{\sigma}$: sample standard deviation; IQR: interquartile range.

Many studies have been published on determining the optimal number of bins [20,26]; however, this discussion is beyond the scope of this study.

Wand [20] stated that practitioners should be cautious about using the default rules of statistical packages in applications where important features in the dataset could be unnoticed. To illustrate, assume a mixed dataset with 99 data from a standard uniform distribution [0:1] and an extreme point valued at 10. For simplicity, we set the $k$ to 10 for both approaches. Figure 1 illustrates both approaches in the presence of an extreme value.

In Figure 1, almost all data in the static bin-width histogram is accumulated in a single bin; hence, this does not represent a reasonable estimation of data. This example indicates why the static bin-width approach would not work well with anomalies. The over-smoothing problem is inevitable if a non-robust rule computes the number of bins in the presence of extreme value(s). Since anomalies are expected in datasets, there is merit in using a robust FD rule for a static bin-width approach. Implementing the FD rule also allows the static bin-width approach to work efficiently with heavy-tailed distributions.

The dynamic bin-width approach is recommended for unknown or heavy-tailed distributions. However, having only one extreme value would yield harmful effects on outlier scores of non-outlier observations. Since the number of data instances is distributed equally to the bins, non-outlier and outliers might fall into the same bin. As evident in Figure 1, having only one extreme value lowered the densities of other instances in the same bin. Eventually, this would increase the false positive rate.

### 2.2. Histogram-Based Outlier Score

HBOS is a non-parametric statistical technique incorporating feature specific densities from univariate histograms. Not only numerical but also categorical data can be used to compute HBOS. Densities for categorical data can be obtained utilizing the frequency of each category. The height of each bin in a histogram represents an estimate of density for each dimension $d$. A static or dynamic method can be used to determine the bin-width. The densities are normalized such that the maximum value is 1. The HBOS of every instance $x$ is calculated using the corresponding density $(p)$ of the bins where the instance is located [15];

$$HBOS(x) = \sum_{i=1}^{d} log\left(\frac{1}{hist_i(p)}\right) \tag{1}$$

an instance is labeled as an anomaly if its score exceeds the predetermined threshold.

HBOS assumes independence of the features, enabling the algorithm to multiply scores derived from inverse densities. As the number of features increases, the drawback of this assumption becomes less critical [7]. Using the logarithm on inverted densities provides robustness against extreme data fluctuations.

## 3. Adjusted Histogram-Based Outlier Score

When working with static bin-width histograms, outliers and inliers may have the lowest densities, which is an additional concern. Since HBOS is a density-based approach, it might not be possible to distinguish these observations density-wise. When the population distribution is unknown, gaps between ordered values can be helpful for detecting outliers [27]. When working with heavy-tailed distributions or in the presence of extreme values, gaps in the histograms tend to occur more frequently. The neighborhoods of the bins and gaps also contain information that could be applied for better density estimation.

With this motivation, we present a novel method based on adjusting bin heights using neighbor bins; the adjustment was calculated by taking the average of the corresponding bin and its adjacent bins' heights. The method uses the FD rule as default to determine the number of bins, enabling the static bin-width histogram to work well with heavy-tailed distributions.

Assuming that a static bin-width histogram consists of $k$ bins, the adjusted height of the $j_{th}$ bin is computed below:

$$h'_j = \frac{h_{j-1} + h_j + h_{j+1}}{3}, \qquad j = 1, \ldots, k \tag{2}$$

where $h_j$ is the height of the $j_{th}$ bin of the histogram.

The prime notation indicates the adjusted version of the corresponding bin. As the first and last bins naturally do not have an adjacent bin, we added dummy bins $h_0$ and $h_{(k+1)}$ with 0 height. The adjusted representation of the bin heights $(h'_j)$ yields the adjusted histogram and these adjusted heights (densities) are subsequently normalized so that the maximum value is 1. Consequently, the AHBOS of every instance x is calculated using the corresponding adjusted height of the bins where the instance is located:

$$AHBOS(x) = \sum_{i=1}^{d} log\left(\frac{1}{Ahist_i(p)}\right) \tag{3}$$

Here, $Ahist$ represents the adjusted histogram, which has been normalized, and $Ahist_i(p)$ stands for the density of the corresponding instance for the *ith* feature. The sum of the logarithm of inverse densities yields the *AHBOS* for each instance.

## 4. Numerical Studies

We conducted real and simulated data applications to evaluate the performance of the AHBOS. We used 17 simulated and 11 real data sets. Receiver operating characteristic (ROC), Precision-Recall (PR) curves, and F-1 score were evaluated as the performance metrics.

The ROC curve is a commonly used metric in binary classification. Provost et al. [28] suggested using the ROC curve over accuracy. The ROC curve represents how the True Positive Rate varies against the False Positive Rate. We used the area under the ROC curve (ROC-AUC) results which is equal to 1 for perfect classification. Naturally, anomalies rarely occur in the data, resulting in imbalanced classification. Davis and Goadrich [29] argued that the performance of the ROC curve is questionable for imbalanced classification. They recommended using the PR curve alongside the ROC curve. The PR curve represents the Precision (P) versus Recall (R) for all thresholds. The area under the PR curve (PR-AUC) ranges from 0 (naive classification) and 1 (perfect classification). We also included the F-1 measure, which yields a balanced result of Precision and Recall: (2×P×R/(P+R)).

For all performance metrics, we applied the Friedman test [30,31] and the Nemenyi post-hoc test [32] to check whether the differences between methods were statistically significant [33]. In the study, the significance level of the tests was set to 0.05.

All computations were performed using "R 4.2" [22]. The "Cutpointr" and "MLmetrics" packages were used to compute ROC-AUC, F-1 score, and PR-AUC [34,35]. Three-dimensional plot was constructed via the "Scatterplot3d" package [36]. Statistical tests were performed with "PMCMRplus" package [37]. R codes are available on https://github.com/eyildiztepe/AHBOS.

### 4.1. Simulation Study

HBOS is based on the assumption of independence between features, and this assumption usually does not hold for real data. Therefore, we generated simulated datasets not only to control global outliers but also to provide gaps. For the sake of simplicity, the number of features was limited to three. We generated data independently from a standard normal distribution (*N(0,1)*), and a certain percentage of data was replaced with outliers. The first two variables contain outliers from shifted chi-square distribution $\left(\chi^2_{(1)} + 5\right)$ and the final variable only contains an extreme value of 100. Figure 2 provides the 3d visualization of one simulation setting when the number of instances is 1000 and the outlier percentage is 0.05. Inliers and outliers are represented as circles and triangles, respectively. Figure 2 clearly illustrates the successful generation of gaps of varying magnitudes between outliers, thereby fulfilling our primary objective in the simulation study. We used a broad range of sample sizes and outlier percentages to cover many possible settings. Also, consistency was acquired by repeating the data generation process ten times, and the results' averages are provided in Table 2. The number of best scores for all settings is indicated at the bottom of the table. Performance metrics were calculated for a predetermined threshold and optimized for the maximum F-1 score. The best results are highlighted in boldface.

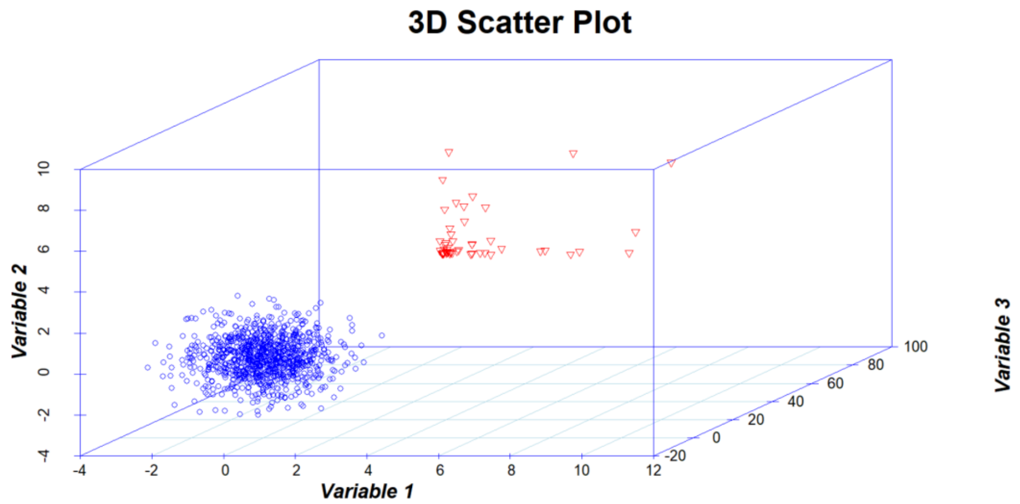


Figure 2. 3d scatter plot of the generated data when # of instances is 1000, and the outlier percentage is 0.05. Circles and triangles indicate normal and anomalous instances, respectively.

Table 2. ROC-AUC, PR-AUC, and F1-Score results for the simulated data

| | | | | *ROC-AUC* | | | *PR-AUC* | | | *F-1 Score* | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ***N*** | ***%*** | ***#*** | ***Sta*** | ***Dyn*** | ***AHBOS*** | ***Sta*** | ***Dyn*** | ***AHBOS*** | ***Sta*** | ***Dyn*** | ***AHBOS*** |
| *100* | *5* | *5* | 0.9938 | 0.9615 | **0.9998** | 0.6906 | 0.3466 | **0.7963** | 0.8812 | 0.6526 | **0.9909** |
| *100* | *10* | *10* | 0.9467 | 0.9566 | **0.9929** | 0.5840 | 0.4891 | **0.8280** | 0.6983 | 0.7814 | **0.9244** |
| *500* | *1* | *5* | 0.9996 | 0.9902 | **1.0000** | 0.7528 | 0.2621 | **0.8000** | 0.9652 | 0.5715 | **1.0000** |
| *500* | *5* | *25* | 0.9847 | 0.9815 | **0.9982** | 0.7509 | 0.5743 | **0.9149** | 0.7332 | 0.7561 | **0.9550** |
| *500* | *10* | *50* | 0.9445 | 0.9411 | **0.9877** | 0.6485 | 0.5214 | **0.8478** | 0.6587 | 0.6897 | **0.8776** |
| *1000* | *1* | *10* | 0.9997 | 0.9964 | **1.0000** | 0.8667 | 0.5117 | **0.8970** | 0.9549 | 0.7666 | **0.9905** |
| *1000* | *5* | *50* | 0.9853 | 0.9900 | **0.9960** | 0.7550 | 0.7285 | **0.8887** | 0.7507 | 0.8053 | **0.8985** |
| *1000* | *10* | *100* | 0.9517 | 0.9350 | **0.9880** | 0.6853 | 0.5506 | **0.8608** | 0.6562 | 0.6452 | **0.8731** |
| *5000* | *1* | *50* | 0.9993 | 0.9993 | **0.9997** | 0.9058 | 0.8249 | **0.9458** | 0.8747 | **0.9523** | 0.9475 |
| *5000* | *5* | *250* | 0.9816 | 0.9857 | **0.9925** | 0.7559 | 0.6970 | **0.8447** | 0.7018 | 0.7463 | **0.7927** |
| *5000* | *10* | *500* | 0.9275 | 0.9004 | **0.9676** | 0.6421 | 0.5070 | **0.7357** | 0.6204 | 0.5688 | **0.6976** |
| *10000* | *1* | *100* | 0.9989 | **0.9996** | 0.9995 | 0.8989 | 0.8933 | **0.9350** | 0.8395 | **0.9734** | 0.8853 |
| *10000* | *5* | *500* | 0.9761 | **0.9877** | 0.9876 | 0.7332 | 0.7232 | **0.7887** | 0.6852 | **0.7786** | 0.7361 |
| *10000* | *10* | *1000* | 0.9293 | 0.9500 | **0.9668** | 0.6402 | 0.6260 | **0.7296** | 0.6123 | 0.6494 | **0.6857** |
| *50000* | *1* | *500* | 0.9990 | 0.9984 | **0.9995** | 0.9070 | 0.8357 | **0.9376** | 0.8399 | 0.7856 | **0.8806** |
| *50000* | *5* | *2500* | 0.9758 | 0.9680 | **0.9875** | 0.7330 | 0.6238 | **0.7893** | 0.6812 | 0.6396 | **0.7296** |
| *50000* | *10* | *5000* | 0.9147 | 0.9066 | **0.9528** | 0.6293 | 0.5774 | **0.6858** | 0.6053 | 0.6170 | **0.6488** |
| ***Best score in*** | | | 0/17 | 2/17 | 15/17 | 0/17 | 0/17 | 17/17 | 0/17 | 3/17 | 14/17 |

Note: *N* is the number of instances. % and # indicate the percentage and number of outliers, respectively. Sta, Dyn, and AHBOS refer to results for static bin-width, dynamic bin-width, and AHBOS, respectively.

ROC-AUC results in the simulation study demonstrated how all methods performed well, and the results were above 0.90. A closer inspection of the ROC-AUC results indicates that AHBOS achieved the maximum scores in 15 of 17 cases, and for the remaining two cases, the results were very close. When considering the PR-AUC results, AHBOS performed best in all cases. Similar to the ROC-AUC results, AHBOS achieved a maximum of 14 of 17 F-1 scores. Dynamic bin-width HBOS yielded the highest F-1 scores in the remaining three cases. The overall results showed that AHBOS is more successful than the other approaches in terms of the performance metrics used in the simulation study. In other words, AHBOS performed best in the presence of gaps and global outliers.

We have used the Friedman test to test the null hypothesis that all methods' performance metrics are equal on the datasets. If the null hypothesis is rejected, Nemenyi post-hoc test is conducted for all pairwise comparisons. The Friedman test and Nemenyi post-hoc test *p-values* are given in Table 3. According to the Friedman test results, *p-values* were lower than the significance level for all metrics. That is, at least one method was different from the others for all metrics. Pairwise comparisons showed that AHBOS is significantly different from the other methods.

Table 3. The Friedman and Nemenyi post-hoc test results for the simulated data

| ***Metric*** | ***Friedman test p-value*** | ***Nemenyi post-hoc test p-value*** | |
|---|---|---|---|
| *ROC-AUC* | **3.97e-5** | Sta vs Dyn | 0.9372 |
| | | AHBOS vs Sta | **0.0007** |
| | | AHBOS vs Dyn | **0.0002** |
| *PR-AUC* | **4.14e-8** | Sta vs Dyn | **0.0010** |
| | | AHBOS vs Sta | **0.0010** |
| | | AHBOS vs Dyn | **1.7e-08** |
| *F-1 Score* | **1.03e-4** | Sta vs Dyn | 0.5585 |
| | | AHBOS vs Sta | **0.0001** |
| | | AHBOS vs Dyn | **0.0057** |

Note: The *p-values* lower than the significance level of 0.05 are given in boldface.

### 4.2. Real Data Studies

There is a vast number of datasets used for anomaly detection in numerous publications, but many have questionable validity, missing references, or are publicly unavailable [38]. For the sake of reliability and reproducibility, we considered some commonly used benchmark datasets. Our real data application comprises 11 benchmark datasets that cover many aspects of the application fields and vary in size, dimension, and outlier percentage. Five data sets were pre-processed datasets published by Goldstein for anomaly detection tasks [39]. The remaining six datasets, which are commonly used benchmark datasets, were obtained from the ODDS Library [40]. A description of the real datasets is provided in Table 4, and the performance metrics are provided in Table 5.

According to ROC-AUC results, out of 11 results, AHBOS had the highest 7, and dynamic bin-width HBOS had the highest 4. Compared with static bin-width HBOS, AHBOS had the highest results or very close values. It can be seen from the PR-AUC and F1-score results that AHBOS performed better than the other methods. AHBOS had 8 best F-1 scores while it was the second in the remaining 3 data sets. The dynamic bin-width HBOS gave better results in all metrics for "annthyroid" dataset. Overall, AHBOS showed better and more competitive performance than other methods, especially for PR-AUC and F1-score.

For real datasets, the Freidman test *p-value was* found 0.075 for ROC-AUC results, and none of the methods were found significantly different from others. The Friedman test *p-values* were found 0.02 and 0.019 for PR-AUC and F-1 score results, respectively. According to Nemenyi post-hoc test results for PR-AUC, AHBOS was significantly different than dynamic bin-width HBOS (*p-value* 0.015). For the F-1 score, the AHBOS and static bin-width HBOS were found significantly different (*p-value* 0.028).

The performance metrics obtained from the real data study agree with the simulation results, although the difference was not as sharp. This can be attributed to the diverse characteristics of the real datasets, such as variations in sample size, number of variables, anomaly percentage, and the presence of gaps.

Table 4. Description of the real datasets

| ***Dataset*** | ***N*** | ***d*** | ***#*** | ***%*** |
|---|---|---|---|---|
| *ionosphere*[2] | 351 | 33 | 126 | 35.90 |
| *b-cancer*[1] | 367 | 30 | 10 | 2.72 |
| *pima*[2] | 768 | 8 | 275 | 35.90 |
| *cardio*[2] | 1831 | 21 | 176 | 9.61 |
| *musk*[2] | 3062 | 166 | 97 | 3.17 |
| *satellite*[1] | 5100 | 36 | 75 | 1.49 |
| *satimage-2*[2] | 5803 | 36 | 71 | 1.22 |
| *pendigits*[2] | 6870 | 16 | 156 | 2.27 |
| *annthyroid*[1] | 6916 | 21 | 250 | 3.61 |
| *shuttle*[1] | 46464 | 9 | 878 | 1.89 |
| *kdd99*[1] | 620098 | 38 | 1052 | 0.17 |

Note: *N* is the number of instances. d, #, and % indicate dimension, number of outliers, and percentage of outliers, respectively. Superscripts on data names give the sources, (1) are from Goldstein [39] and (2) are from ODDS Library [40].

Table 5. ROC-AUC, PR-AUC, and F1-Score results for real datasets

| ***Dataset*** | ***ROC-AUC*** | | | ***PR-AUC*** | | | ***F-1 Score*** | | |
|---|---|---|---|---|---|---|---|---|---|
| | ***Sta*** | ***Dyn*** | ***AHBOS*** | ***Sta*** | ***Dyn*** | ***AHBOS*** | ***Sta*** | ***Dyn*** | ***AHBOS*** |
| *ionosphere*[2] | 0.5224 | **0.6927** | 0.6912 | 0.3140 | 0.2575 | **0.4780** | 0.5768 | 0.5967 | **0.6560** |
| *b-cancer*[1] | **0.9826** | 0.9801 | 0.9824 | **0.5522** | 0.4225 | 0.5502 | **0.6250** | 0.5556 | **0.6250** |
| *pima*[2] | 0.6402 | 0.6544 | **0.6738** | 0.5096 | 0.5077 | **0.5238** | 0.5291 | 0.5451 | **0.5577** |
| *cardio*[2] | 0.8514 | 0.8400 | **0.8986** | 0.4741 | 0.3895 | **0.5117** | 0.4850 | 0.4696 | **0.5391** |
| *musk*[2] | 0.9999 | **1.0000** | **1.0000** | 0.9874 | **0.9897** | 0.9895 | 0.9848 | **1.0000** | 0.9949 |
| *satellite*[1] | 0.9232 | **0.9294** | 0.9291 | 0.5801 | 0.5657 | **0.6177** | 0.5970 | 0.6074 | **0.6143** |
| *satimage-2*[2] | 0.9816 | 0.9804 | **0.9850** | 0.7873 | 0.7992 | **0.8250** | 0.7737 | 0.8000 | **0.8085** |
| *pendigits*[2] | 0.9356 | 0.7875 | **0.9400** | 0.2673 | 0.1573 | **0.2699** | 0.3486 | 0.2576 | **0.3655** |
| *annthyroid*[1] | 0.8323 | **0.9160** | 0.8508 | 0.1605 | **0.2812** | 0.1809 | 0.2645 | **0.3311** | 0.2772 |
| *shuttle*[1] | **0.9966** | 0.9933 | **0.9966** | **0.8782** | 0.7640 | 0.8754 | **0.8568** | 0.7115 | 0.8547 |
| *kdd99*[1] | 0.9979 | 0.9936 | **0.9981** | 0.7056 | 0.6772 | **0.7185** | 0.6678 | 0.6600 | **0.6813** |
| ***Best score in*** | 2/11 | 4/11 | 7/11 | 2/11 | 2/11 | 7/11 | 2/11 | 2/11 | 8/11 |

Note: Sta, Dyn, and AHBOS refer to results for static bin-width, dynamic bin-width, and AHBOS, respectively. Superscripts on data names give the sources, (1) are from Goldstein [39] and (2) are from ODDS Library [40].

## 5. Conclusion

In the Internet era, it is vital to take prompt and immediate action in situations such as credit card fraud detection, intrusion detection, and enemy activity surveillance. The ever-increasing amount of data and the expectation of rapid response in modern applications has greatly increased the need for fast and accurate methods. HBOS meets the expectations since it is a quick unsupervised anomaly detection method and can be calculated using either a static or dynamic bin-width histogram. HBOS has gained substantial popularity and is widely used in many application areas and benchmark studies [8,17,18].

However, several issues directly affect the performance of the HBOS. The choice of incorrect bin number causes an over-smoothing problem in the static bin-width approach in the presence of extreme values. This explains why the dynamic bin-width approach has gained popularity over the static bin-width approach. However, the dynamic bin-width approach tends to label inliers as anomalies in the presence of gaps. This issue can be fixed in the static bin-width approach by implementing a commonly used robust bin size rule FD. Since the FD rule is robust to extreme values, the constructed histograms are not affected by extreme values. Another issue about the static bin-width approach is that inliers and outliers might get equal densities. In this case, histogram gaps could be used to remedy this issue. Gaps in the static bin-width histogram are more likely to occur in circumstances when the parent distribution has a heavy tail or anomalies exist. However, scores obtained from static bin-width histograms do not consider the gaps that could be used to increase the performance of HBOS. This study presents the AHBOS, which modifies the static bin-width histogram's densities depending on both the bins and their neighbors. Simulation and real data application results supported the fact that the proposed method improves performance. Based on the simulation results, the AHBOS performed better than the HBOS method for all performance metrics, especially for the PR-AUC measure. Except for a few cases, the real data study also yielded a result supporting the simulation outcomes. The new approach demonstrated that it could compete well with dynamic bin-width HBOS when dealing with heavy-tailed or unknown distributions. The adjustment also performed well in the absence of gaps. Furthermore, AHBOS can be improved through the implementation of various algorithms for determining the bin number. In future work, the performance of the AHBOS will be investigated in intrusion detection applications, where HBOS is widely used. In conclusion, the AHBOS holds promise for application in both research and practical scenarios, overcoming the limitations of traditional HBOS methods.

## 6. References

[1] Chandola, V., Banerjee, A., and Kumar, V., "Anomaly Detection: a Survey", *ACM Computing Surveys (CSUR),* 41(3), 1-58, 2009.

[2] Anscombe, F. J., "Rejection of Outliers", *Technometrics,* 2(2), 123-146, 1960.

[3] Grubbs, F. E., "Procedures for Detecting Outlying Observations in Sample", *Technometrics*, 11(1), 1-21, 1969.

[4] Hawkins, D. M., *Identification of Outliers,* London: Chapman and Hall, 1980.

[5] Breunig, M. M., Kriegel, H. P., Ng, R. T. and Sander, J., "LOF: Identifying Density Based Local Outlier", *In Proceedings of the 2000 ACM SIGMOD International Conference on Management of data*, 2000, 93-104.

[6] Hodge, V. and Austin, J., "A survey of Outlier Detection methodologies", *Artificial Intelligence Review*, 22, 85-126, 2004.

[7] Goldstein, M. and Uchida, S., "A Comparative Evaluation of Unsupervised Anomaly Detection Algorithms for Multivariate Data", *PloS One*, 11(4), 2016.
[8] Zoppi, T., Ceccarelli, A., Puccetti, T. and Bondavalli, A., "Which Algorithm Can Detect Unknown Attacks? Comparison of Supervised, Unsupervised and Meta-Learning Algorithms for Intrusion Detection", *Computers & Security*, 127, 2023.
[9] Kind, A., Stoecklin, M. P. and Dimitropoulos, X., "Histogram-Based Traffic Anomaly Detection", *IEEE Transactions on Network and Service Management*, 6(2), 110-121, 2009.
[10] Sabau, A. S., "Survey of Clustering Based Financial Fraud Detection Research", *Informatica Economica*, 16(1), 110, 2012.
[11] Xie, M., Hu, J. and Tian, B., "Histogram-Based Online Anomaly Detection in Hierarchical Wireless Sensor Network", *In 2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing And Communications*, 2012. 751-759.
[12] Sharma, A., Pujari, A. K. and Paliwal, K. K., "Intrusion Detection Using Text Processing Techniques with a Kernel Based Similarity Measure", *Computers & Security*, 26(7-8), 488-495, 2007.
[13] Carminati, M., Polino, M., Continella, A., Lanzi, A., Maggi, F. and Zanero, S., "Security Evaluation of a Banking Fraud Analysis System", *ACM Transactions on Privacy and Security (TOPS)*, 21(3), 1-31, 2018.
[14] Munir, M., Chattha, M. A., Dengel, A. and Ahmed, S., "A Comparative Analysis of Traditional and Deep Learning-Based Anomaly Detection Methods For Streaming Data", *In 2019 18th IEEE International Conference On Machine Learning and Applications (ICMLA),* 2019, 561-566.
[15] Goldstein, M. and Dengel, A., "Histogram-Based Outlier Score (HBOS): A Fast Unsupervised Anomaly Detection Algorithm", *KI-2012: Poster and Demo Track*, 59-63, 2012.
[16] Saba-Sadiya, S., Chantland, E., Alhanai, T., Liu, T. and Ghassemi, M. M., "Unsupervised EEG Artifact Detection and Correction", *Frontiers in Digital Health*, 2, 2021.
[17] Han, S., Hu, X., Huang, H., Jiang, M. and Zhao, Y., "Adbench: Anomaly Detection Benchmark.", *Advances in Neural Information Processing Systems*, 35, 32142-32159, 2022.
[18] Dobos, D., Nguyen, T. T., Dang, T., Wilson, A., Corbett, H., McCall, J. and Stockton, P., "A Comparative Study of Anomaly Detection Methods for Gross Error Detection Problems", *Computers & Chemical Engineering*, 175, 2023.
[19] Paulauskas, N. and Baskys, A., "Application of Histogram-Based Outlier Scores to Detect Computer Network Anomalies", *Electronics*, 8(11), 1251, 2019.
[20] Wand, M. P., "Data-Based Choice of Histogram Bin Width", *The American Statistician*, 51(1), 59-64, 1997.
[21] Sturges, H. A., "The Choice of a Class Interval", *Journal of the American Statistical Association*, 21(153), 65-66, 1926.
[22] R Core Team, *R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/*, 2022.
[23] Scott, D. W., "On Optimal and Data-Based Histograms", *Biometrika*, 66(3), 605-610, 1979.
[24] Freedman, D. and Diaconis, P., "On the Histogram as a Density Estimator: $L_2$ Theory", *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, 57(4), 453-476, 1981.
[25] Venables, W. N. and Ripley, B. D., *In Modern Applied Statistics with S*, Springer, New York, 2002.
[26] Shimazaki, H. and Shinomoto, S., "A Method for Selecting the Bin Size of a Time Histogram", *Neural Computation*, 19(6), 1503-1527, 2007.
[27] Wilkinson, L., "Visualizing Big Data Outliers Through Distributed Aggregation", *IEEE Transactions on Visualization and Computer Graphics*, 24(1), 256-266, 2017.
[28] Provost, F. J., Fawcett, T. and Kohavi, R., "The Case Against Accuracy Estimation for Comparing Induction Algorithms", *In ICML*, 1998, 445-453.
[29] Davis, J. and Goadrich, M., "The Relationship Between Precision-Recall and ROC Curves", *In Proceedings of the 23rd International Conference on Machine Learning*, 2006, 233-240.
[30] Friedman, M., "The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance", *Journal of the American Statistical Association*, 32(200), 675-701, 1937.
[31] Friedman, M., "A Comparison of Alternative Tests of Significance for the Problem of M Rankings", *The Annals of Mathematical Statistics*, 11(1), 86-92, 1940.
[32] Nemenyi, P. B., *Distribution-Free Multiple Comparisons*, PhD Thesis, Princeton University, 1963.
[33] Demšar, J., "Statistical Comparisons of Classifiers Over Multiple Data Sets", *The Journal of Machine Learning Research*, 7, 1-30, 2006.
[34] Thiele, C. and Hirschfeld, G.,"Cutpointr: Improved Estimation and Validation of Optimal Cutpoints In R.", *Journal of Statistical Software*, 98(11), 1-27, 2021.
[35] Yan, Y., *MLmetrics: Machine Learning Evaluation Metrics. R package version 1.1.1*, 2016.
[36] Ligges, U. and Mächler, M., *Scatterplot3d an R Package for Visualizing Multivariate Data*. Technical Report, 2002.
[37] Pohlert, T., *PMCMRplus: Calculate Pairwise Multiple Comparisons of Mean Rank Sums Extended*, 2022.
[38] Campos, G. O., Zimek, A., Sander, J., Campello, R. J., Micenková, B., Schubert, E., Assent, I. and Houle, M. E., "On the Evaluation of Unsupervised Outlier Detection: Measures, Datasets, and an Empirical

Study", *Data Mining and Knowledge Discovery*, 30(4), 891-927, 2016.

[39] Goldstein, M., Unsupervised Anomaly Detection Benchmark. Harvard Dataverse, 2015. doi: 10.7910/DVN/OPQMVF.

[40] Rayana, S., ODDS Library [http://odds.cs.stonybrook.edu]. Stony Brook, NY: Stony Brook University, Department of Computer Science, 2016.