DeSAD: Density Clustering-Guided Streaming Data Anomaly Detection

Yue Zhang[1], Xuchuang Ding[1], Xiang Zhang[2], Wei Ai[2], Gengwen Huang[2], and Yiqun Zhang[3](✉)

[1] Guangdong Polytechnic Normal University, Guangzhou, Guangdong, China
zhangyue@gpnu.edu.cn
[2] Shenzhen BH-Energy Technology Co., Ltd., Shenzhen, Guangdong, China
[3] Guangdong University of Technology, Guangzhou, Guangdong, China
yqzhang@gdut.edu.cn

Abstract. With the rapid increase in Internet of Things, large-scale streaming data is being generated at an unprecedented rate. Identifying and handling anomalies in these data streams is crucial for improving data quality and enhancing data analysis performance. Clustering is an effective method for processing streaming data, as it groups data points to uncover patterns. This article introduces Density clustering-guided Streaming data Anomaly Detection (DeSAD) algorithm, which achieves efficient anomaly detection by dynamically learning density-based clusters as reference sets while eliminating the need for manual parameter tuning. DeSAD first uses density-based clustering techniques to identify potential noise points, and then filters out the true anomalies from these points. This process not only increases computational efficiency but also adapts to the diversity and dynamic nature of streaming data. In addition, DeSAD adjusts its clustering parameters in real-time based on the data distribution, making it highly suitable for varying types of data streams. Experimental results demonstrate its effectiveness in detecting anomalies across 11 benchmark datasets and 4 synthetic datasets, showcasing its potential in handling large-scale, dynamic data streams.

Keywords: Stream data · Anomaly detection · Density-based measure

1 Introduction

With the rapid advancement of digital technologies and electronic manufacturing, data streams are now continuously generated from various facets of human activities, including industrial processes, financial transactions, and online interactions. As a result, efficiently and accurately identifying anomalous patterns in such complex and dynamic systems has become a significant challenge [1] and is a necessary data preprocessing stage for many important unsupervised learning tasks, including missing value imputation [2], clustering [3–5], concept drift detection [6, 7], etc. The high volume and velocity of incoming data, often arriving in large quantities within short time-frames, can quickly lead to substantial data accumulation if not processed in a timely manner

D.-S. Huang et al. (Eds.): ICIC 2025, LNCS 15850, pp. 201–212, 2025.
https://doi.org/10.1007/978-981-96-9884-4_17

[8]. This creates a range of distinctive challenges for machine learning algorithms [9, 10]. One of the most persistent issues in streaming anomaly detection is the difficulty of reliably distinguishing true anomalies from background noise or minor fluctuations.

A conventional streaming data anomaly detection method called xStream is developed in [11]. ECOD [12] and ROD [13] introduce advances as parameter-free. AADS [14] is an autonomous, online anomaly detection and clustering algorithm. Streaming TEDA [15] is optimized for real-time, hardware-based detection. DIF [16] enhances accuracy using neural networks and random cuts. LODA [17] is an ensemble method for fast and robust anomaly detection. When tested on larger datasets with a small proportion of anomalies, the performance of the above methods deteriorates. Density-based clustering [18–20] identifies clustering structures based on sample distribution tightness. They focus on sample density and expand clusters by connecting samples, ultimately producing the final results, and leveraging data correlations. Since they offer several advantages: they detect clusters of arbitrary shapes, are effective for large-scale datasets, and adapt well to high-dimensional data, the density-based method is ideal for analyzing complex, large-scale, and high-dimensional data [21].

To eliminate the need for manual parameter setting of the density-based methods, the k-nearest neighbors (KNN) method [22] is usually employed. Distances to the k-th nearest neighbor [23] are computed, and the average of these distances is used as the epsilon parameter for DBSCAN [24]. Although the parameter k in KNN still requires manual specification, the concept of natural neighbors [25] is utilized to determine an appropriate value for k. The minPoints parameter is dynamically set based on the specific range of natural neighbor feature values in the dataset. As a result, all these processes avoid manual parameter tuning. To address the issue of insensitivity to noise in density-based clustering, anomaly scores are introduced to distinguish between potential and actual anomalies [26].

Based on the above advances, this paper proposes DeSAD, a Density-based clustering algorithm for Streaming data Anomaly Detection, which adopts sliding windows to organize streaming data. The main contributions are efficient anomaly detection through clustering and a sliding window approach. In summary, the contributions of this paper are:

1. DeSAD is a novel, density-based anomaly detection model that outperforms existing methods by dynamically adjusting clustering parameters and preserving historical data, enhancing noise and anomaly detection.
2. The method incorporates clustered natural neighbour relationships and cluster-guided anomaly scoring to efficiently compute real-time anomaly scores, ensuring robust, parameter-free performance.
3. DeSAD is highly interpretable and suitable for lightweight, real-time applications, where limited computational resources and high interpretability are crucial.

2 Related Work

Anomaly detection algorithms can be broadly classified into offline and online approaches. Offline methods, such as ROD [13], ECOD [12] and GMM [27], typically require access to the full dataset and are sometimes adapted for streaming scenarios.

In contrast, online methods like xStream [11], LODA [17], DIF [16] and AADS [14] process data incrementally, making them suitable for real-time, dynamic environments.

In terms of modeling assumptions, one group of methods relies on statistical distribution models. For example, ECOD uses empirical distributions to detect extreme values, while GMM assumes data originates from a mixture of Gaussian distributions. These methods are theoretically grounded and perform well on stable distributions, but they struggle with high-dimensional, multimodal, or non-Gaussian data, and often require prior assumptions or parameter tuning, limiting their robustness. Another group, including LODA, xStream, and ROD, focuses on data structure and sparsity. These methods use techniques like projections, cuts, rotations, or complexity estimation to assign anomaly scores, without assuming specific data distributions. They are generally more flexible and adaptable, and some of them, like LODA and xStream, support online detection. However, they often depend on hyperparameter settings and may scale poorly in high-dimensional spaces. Their interpretability is also often limited due to ensemble or randomized mechanisms. Methods like Streaming TEDA [15] and AADS integrate online learning with streaming frameworks using sliding windows [28, 29], parallel processing, and adaptive clustering [30]. They offer low-latency, efficient performance, especially in resource-constrained environments. Nevertheless, they often rely on sensitive thresholds and may react slowly to abrupt data changes.

Overall, anomaly detection methods involve trade-offs between distribution assumptions, efficiency, parameter sensitivity, and scalability. Developing models that are accurate, adaptive, and low in dependency on prior knowledge remains a key challenge for complex streaming data applications.

3 Proposed Method

3.1 Normal and Anomaly Sample Pre-partition

Density-based clustering is first applied to divide the data into two categories: normal points and potential anomalies. The nearest neighbor method introduced in [25] is utilized to enable parameter-free input for the two parameters of DBSCAN. For the k value in k-nearest neighbors, the value r derived from the concept of natural neighbors is adopted. This r corresponds to the desired k.

For each point i in the dataset, the distances to all other points are first calculated. The distance matrix D is denoted as $\{d_{ij}\}$, where d_{ij} represents the distance between point i and point j. These distances are then sorted in ascending order to form a list of distances from point i to all other points. The k-th smallest distance, $d_i^{(k)}$, is subsequently selected, representing the distance from point i to its k-th nearest neighbor. This is mathematically expressed as:

$$d_i^{(k)} = \text{sorted}(d_i)[k], \tag{1}$$

where $\text{sorted}(d_{i1}, d_{i2}, \ldots, d_{in})$ denotes the list of distances from point i to all other points sorted in ascending order, and the subscript k refers to the k-th smallest value in this sorted list.

Let $S = \{1, 2, \ldots, N\}$ be the dataset, where N is the total number of points. For each point i, the k-th nearest distance, denoted as $\text{sorted}(d_i)_k$, is the distance from point i to its k-th closest neighbor after sorting the distances to all other points. Next, the average value ϵ of the k-th nearest neighbor distances for all points in the dataset is computed. This average ϵ is given by:

$$\in = \frac{1}{N} \sum_{i=1}^{N} \text{sotred}(d_i)[k]. \tag{2}$$

Here, N is the total number of points, and $\text{sorted}(d_i)_k$ is the k-th nearest distance from point i to its k-th closest neighbor. The average value ϵ serves as the DBSCAN epsilon, defining the neighborhood radius. By averaging the k-th nearest distances, the dataset's density is captured, allowing core points (with many neighbors) to be distinguished from noise points (isolated ones). This ϵ helps fine-tune DBSCAN's density threshold for better clustering.

As mentioned in the paper [25], "the value range of λ must be $2 \leq \lambda < n$, and it is generally 6 or 7. For high-dimensional or irregular datasets, λ will be more than 20 but less than 30." Since the value of r varies across different datasets, another parameter of DBSCAN, minPoints, is dynamically set, as shown in:

$$\text{minPoints}(r) = \begin{cases} 2r & \text{if } r \leq 20 \\ r & \text{if } 20 < r \leq 30 \\ \left\lceil \frac{r}{2} \right\rceil & \text{if } r > 30. \end{cases} \tag{3}$$

The two DBSCAN parameters are determined, and clustering is performed within each window, dividing the data into clusters and noise points (potential anomalies). The noise points are then identified as anomalies in the final detection stage.

3.2 Fine Anomaly Detection of Streaming Data

Anomaly detection is a critical stage in the proposed DeSAD algorithm. This stage introduces anomaly scores based on the formed clustering results. Points with anomaly scores greater than the threshold are declared as anomalies, while those below are considered normal. Based on the clustering results, the mean values of the centroids of normal clusters:

$$C = \frac{1}{|L|} \sum\nolimits_{l \in L} \left(\frac{1}{|S_l|} \sum\nolimits_{X_{i \in S_l}} x_i \right), \tag{4}$$

will be utilized, where L is the set of normal cluster labels, representing all the labels of normal clusters, S_l is the set of data points in cluster l, containing all the points in cluster l, $|S_l|$ is the number of data points in cluster l, $|L|$ is the number of normal clusters. Then, the Euclidean distance from each noise point to the centroid mean value is calculated as:

$$d = \{||x_i - C||(x_i \in M)\}. \tag{5}$$

Here, M denotes the set of noise points, and $||x_i - C||$ is the Euclidean distance between a noise point x_i and the centroid mean C. This distance reflects how far a noise point

deviates from the center of normal clusters. The value d in Eq. (5) is used as the anomaly score.

Finally, the threshold y is defined as the average distance from all points to the centroid mean, as expressed by the following equation:

$$y = \frac{1}{|M|} \sum_{X_i} ||X_i - C||, \tag{6}$$

where |M| is the number of points in the set M (i.e., the noise points). The threshold y is used to determine which noise points have a distance greater than this threshold, thus being considered as anomalies. By comparing the distances to this threshold, outliers in the dataset can be effectively identified.

Anomalies are detected by comparing data point scores in each sliding window to a threshold. Those exceeding it are labeled as anomalies. The algorithm processes the data stream window by window, performing clustering, calculating centroids of normal points, and updating scores. A half-window sliding step preserves boundary information, enabling continuous adaptation to data changes. The time complexity of DeSAD is dominated by data loading, feature computation, DBSCAN clustering, and sliding window operations, with a worst-case estimate of $O(n \times d + m \times r \times W \log W)$, where n, d, m, r, and w represent dataset size, feature dimension, number of windows, neighborhood radius, and window size, respectively.

4 Experiments

4.1 Performance Comparison

The performance of the proposed DeSAD is compared with eight other algorithms on 15 datasets (statistics shown in Table 1). As shown in Fig. 1, both our method and AADS perform well on smaller datasets such as Breastw [31], Spect [14], Parkinsons [32], and Pima [33]. However, on larger, higher-dimensional datasets like Satimage [34], Optdigits [35], Cardio [36], and Letter [37], AADS's performance drops, while ours maintains high accuracy. AADS struggles with high-dimensional data, while our DeSAD utilizes a parameter-free natural neighbor method and excels in anomaly detection.

The complete results are shown in Table 2, with the best and second-best results highlighted in bold and underlined, respectively. DeSAD ranks first (1.5556), far surpassing the other methods (the second-best, TEDA, ranks 3.1333). TEDA stands out particularly in terms of Recall rate, indicating its excellent performance in identifying anomalies. DeSAD also performs exceptionally well in Recall rate, ranking among the top two on many datasets, demonstrating its success in detecting the majority of true anomalies with very few missed detections. For the synthetic dataset D1 with concentrated normal data, most methods performed well, except xStream, TEDA, and DIF, while our method achieved the best results. As the number of points in D1 increased and the anomaly ratio in D2 decreased, the performance of ROD, ECOD, and GMM declined significantly, whereas our method demonstrates robustness. On datasets with less distinguishable anomalies (D3 and D4), the performance of xStream and TEDA remained limited, while DIF exhibited some improvement. ECOD, ROD, LODA, and

GMM showed a noticeable decline in performance. In contrast, our method consistently maintained high accuracy. Overall, DeSAD delivered the best performance across all experiments.

Table 1. Statistics of the 15 datasets. The columns "N" and "Outliers" represent the number of samples and the number of outliers, respectively.

Category	Datasets	N	Outliers	Category	Datasets	N	Outliers
Real	Breastw	683	239	Real	Pima	768	268
Real	Spect	267	212	Real	Ecoli	336	64
Real	Optdigits	5216	150	Real	Wbc	378	21
Real	Satimage	5803	71	Synthetic	D1	1000	100
Real	Letter	1600	100	Synthetic	D2	1450	100
Real	Cardio	1831	176	Synthetic	D3	2000	100
Real	Parkinsons	195	147	Synthetic	D4	2950	100
Real	Vertebral	240	30	-			

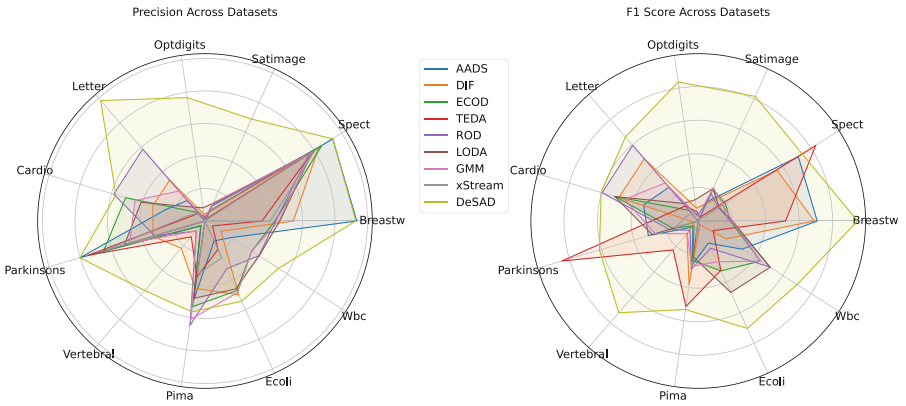

Fig. 1. Performance of various anomaly detection methods across different datasets under Precision (P) and F1 Score.

Moreover, we evaluated the average execution time of DeSAD in comparison with four other online anomaly detection algorithms, i.e., AADS, LODA, xStream, and DIF, across eight datasets. As shown in Table 3, DeSAD achieves the highest efficiency, ranking first among the five evaluated online algorithms.

4.2 Visualization Results of Synthetic Data

To evaluate the effectiveness of the DeSAD algorithm, we visualized the anomaly detection results on synthetic datasets. Since D1 and D2 are the same type of data, differing

Table 2. Performance metrics comparison across 11 real and 4 synthetic datasets.

Datasets	Metrics	AADS	DIF	ECOD	TEDA	ROD	LODA	GMM	xStream	DeSAD
Breastw	P	0.8645	0.5431	0.4303	0.3510	0.4352	0.4612	0.4537	0.4014	**0.9280**
	F1	0.7156	0.6910	0.1911	0.5199	0.1948	0.2043	0.2017	0.1798	**0.9490**
	RR	0.6145	0.9494	0.1229	**0.9996**	0.1255	0.1313	0.1298	0.1159	0.9707
Spect	P	0.9342	0.8318	0.8520	0.7864	0.7632	0.7845	0.7817	0.8049	**0.9348**
	F1	0.7085	0.5580	0.1924	0.8313	0.1443	0.1483	0.1765	0.1754	**0.7371**
	RR	0.5717	0.4198	0.1084	**0.8821**	0.0811	0.0832	0.0995	0.0811	0.6085
Satimage	P	0.0680	0.0600	0.1102	0.0124	0.1015	0.1188	0.1222	0.0030	**0.6893**
	F1	0.1272	0.1131	0.2055	0.0242	0.1810	0.2117	0.2178	0.0050	**0.8161**
	RR	0.9718	0.9975	0.9014	**0.9998**	0.8310	0.9718	0.9956	0.0141	0.9986
Optdigits	P	0.0910	0.0420	0.0000	0.0280	0.0000	0.0824	0.0172	0.0159	**0.7667**
	F1	0.0330	0.0780	0.0000	0.0560	0.0000	0.1280	0.0268	0.0020	**0.8366**
	RR	0.1264	0.5400	0.0000	**0.9987**	0.0000	0.2867	0.0600	0.0267	0.9200
Letter	P	0.1671	0.3320	0.0688	0.0626	0.5847	0.1125	0.2437	0.0658	**0.9804**
	F1	0.2667	0.4903	0.0846	0.1179	0.5961	0.1385	0.3000	0.0568	**0.6623**
	RR	0.6500	0.9375	0.1100	**0.9961**	0.6080	0.1800	0.3900	0.0500	0.5000
Cardio	P	0.2507	0.3347	0.5082	0.0962	0.5847	0.1125	0.2437	0.0658	**0.9804**
	F1	0.3400	0.4933	0.5181	0.1756	0.5961	0.5181	0.4568	0.0803	**0.6022**
	RR	0.5284	0.9375	0.5284	**0.9921**	0.6080	0.4261	0.4659	0.0625	0.6364
Parkinsons	P	0.8000	0.3333	0.6500	0.7487	0.3226	0.5161	0.4839	0.6316	**0.8021**
	F1	0.3094	0.0238	0.1557	**0.8462**	0.1786	0.2857	0.2679	0.1455	0.6134
	RR	0.1918	0.0123	0.0884	**0.9728**	0.1235	0.1975	0.1852	0.1235	0.4965
Vertebral	P	0.0370	0.2222	0.0417	0.1288	0.0833	0.0000	0.0800	0.0000	**0.5660**
	F1	0.0476	0.1026	0.0370	0.2281	0.0741	0.0000	0.1000	0.0000	**0.7229**
	RR	0.0667	0.0667	0.0333	0.9846	0.0667	0.0000	0.1300	0.0000	**0.9932**
Pima	P	0.4949	0.4185	0.5325	0.3473	**0.6494**	0.4805	0.6104	0.3330	0.5649
	F1	0.2678	0.3838	0.2377	0.5146	0.2899	0.2145	0.2725	0.1016	**0.5325**
	RR	0.1835	0.3545	0.1530	**0.9925**	0.1866	0.1381	0.1754	0.0599	0.5037
Ecoli	P	0.1351	0.5000	0.4706	0.1928	0.3226	0.4590	0.4839	0.2500	**0.5428**
	F1	0.1449	0.0241	0.3265	0.3232	0.1786	0.4680	0.2679	0.0294	**0.7033**
	RR	0.1562	0.0123	0.2500	0.9908	0.1235	0.4773	0.1852	0.0156	**0.9924**
Wbc	P	0.1895	0.1171	0.3421	0.0560	0.3947	0.3950	0.3421	0.0000	**0.5385**
	F1	0.3103	0.1970	0.4407	0.1061	0.5085	0.5090	0.4407	0.0000	**0.7000**
	RR	0.8571	0.6190	0.6350	0.9936	0.7143	0.7143	0.6190	0.0000	**0.9986**
D1	P	0.7869	0.5272	0.9400	0.1015	0.8900	0.9184	0.9600	0.0857	**0.9896**
	F1	0.8649	0.6831	0.9401	0.1842	0.8900	0.9091	0.9600	0.0444	**0.9694**
	RR	0.9600	0.9700	0.9400	0.9900	0.8900	0.9000	0.9600	0.0300	**0.9923**

(*continued*)

Table 2. (*continued*)

Datasets	Metrics	AADS	DIF	ECOD	TEDA	ROD	LODA	GMM	xStream	DeSAD
D2	P	0.4026	**1.0000**	0.6414	0.0704	0.5862	0.4762	0.6345	0.0000	**1.0000**
	F1	0.5619	0.8950	0.7592	0.1315	0.6939	0.5310	0.7510	0.0000	**0.9134**
	RR	0.9800	**1.0000**	0.9900	**1.0000**	0.9200	0.9800	**1.0000**	0.0400	0.9986
D3	P	0.7840	0.3390	0.4950	0.0698	0.4600	0.7424	0.5000	0.0465	**0.9892**
	F1	0.8711	0.5063	0.6600	0.1305	0.6133	0.8448	0.6667	0.0430	**0.9534**
	RR	0.9300	0.8100	0.9300	**1.0000**	0.8500	0.6000	0.9200	0.0000	0.8700
D4	P	0.2043	**1.0000**	0.3220	0.0343	0.2475	0.1797	0.3085	0.0877	**1.0000**
	F1	0.3357	0.9011	0.4810	0.0064	0.3696	0.2684	0.4608	0.0809	**0.9305**
	RR	0.9400	0.8200	0.9500	**1.0000**	0.7300	0.5300	0.9100	0.0750	0.8700
Avg.Rank		3.7333	3.8222	4.6889	3.1333	4.5333	4.6667	3.8222	6.7333	**1.5556**

Table 3. Execution time comparison (in seconds)

Algorithms	Breastw	Spect	Satimage	Optdigits	Letter	Cardio	Parkinsons	Vertebral
DeSAD	0.2062	0.4418	0.1207	0.1611	0.1180	0.1291	0.1907	0.2226
AADS	0.9864	0.3724	8.9657	7.3955	2.2316	2.5469	0.3013	0.3103
DIF	1.1858	0.5749	9.3869	8.1955	2.5050	0.8987	0.5867	0.4041
LODA	8.0635	3.0661	66.638	59.8980	18.3736	20.7038	2.2049	2.8333
xStream	57.9340	25.9208	63.5481	63.8158	63.8673	61.3182	62.4238	59.8425

only in quantity, and the same applies to D3 and D4, we chose D1 and D3 as representative datasets for visualization.

As illustrated in Fig. 2, our method demonstrated a high level of performance in identifying anomaly points in the D1 dataset, with almost all anomalies detected and a very low false positive rate. The algorithm was able to clearly differentiate between normal and anomalous data points, effectively identifying outliers while minimizing misclassifications. On the D3 dataset, also known as the "smiley face" dataset, the situation was more challenging. In this case, the anomaly points and normal points are highly similar, making it difficult to distinguish between the two categories. Despite this, the DeSAD algorithm still managed to identify the majority of the anomalies. While a few normal points were incorrectly labeled as anomalies and some anomalies went undetected, the overall performance remained impressive, as most of the anomalies were successfully detected, confirming the algorithm's robustness in handling more complex and subtle patterns in the data.

4.3 Parameter Sensitivity Evaluation

In the experiments, a sliding window technique was used to process the data stream, and the DeSAD algorithm's anomaly detection accuracy was evaluated under varying window sizes. Window sizes were adjusted based on each dataset to test the method's

stability. Using the Breastw and Letter datasets, precision and F1 scores were plotted across different window sizes as shown in Fig. 3. Results showed that DeSAD maintained consistently high performance, with minimal variation in both metrics. This indicates strong stability and effectiveness under different parameter settings, confirming DeSAD's suitability for diverse and dynamic data stream environments.

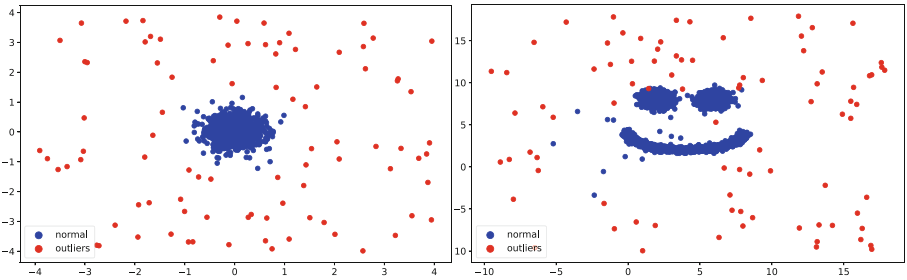

Fig. 2. Visualization of synthetic datasets D1 (left) and D3 (right). Normal samples and outliers are marked in blue and red, respectively. (Color figure online)

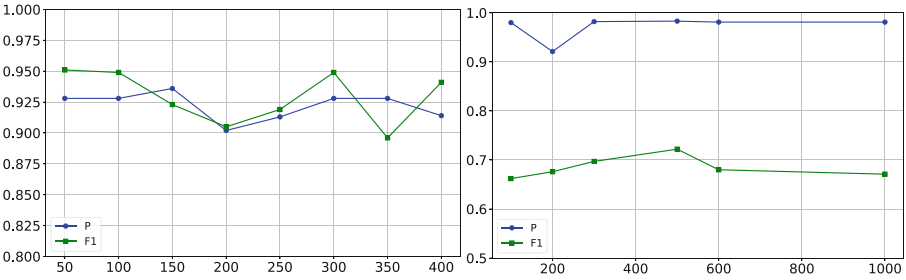

Fig. 3. The P and F1 performance on the datasets Breastw (left) and Letter (right), under varying window configurations. The y-axis of the figure represents the values of the performance metrics P and F1, while the x-axis denotes different window sizes.

4.4 Ablation Study

Previous methods treated all clustering noise as anomalies. DeSAD introduces anomaly scores to better distinguish noise from true outliers. The method without incorporating anomaly scores is referred to as NoScore. By comparing the algorithm's performance before and after incorporating anomaly scores, it highlights the significant performance improvement achieved by the DeSAD method. Figure 4 compares DeSAD and NoScore on real and synthetic datasets using heatmaps. On real datasets, DeSAD achieves higher precision (e.g., 0.9804 on Optdigits) and F1 score (e.g., 0.9490 on Breastw) than NoScore, indicating better performance. On synthetic datasets, DeSAD reaches perfect precision (1.0000) and F1 scores above 0.9000, while NoScore shows low precision and F1. Overall, DeSAD is more reliable and effective for anomaly detection.

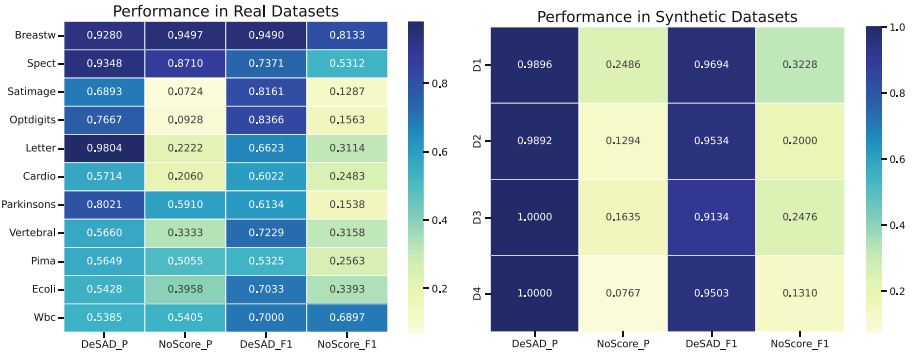

Fig. 4. The heatmaps show the performance of DeSAD and NoScore on real and synthetic datasets, comparing Precision (P) and F1 Score values. The vertical axis of the heatmap represents the datasets, while the horizontal axis represents the performance before and after the ablation study.

5 Concluding Remarks

This paper presents DeSAD, a stream data anomaly detection algorithm. It clusters input data to identify prominent distributions and compute anomaly scores, using density measures to distinguish intra-cluster and scattered objects. Clusters and scores are updated within each new data chunk for continual anomaly detection. Experimental results show that DeSAD outperforms competitors, remains stable with varying window sizes. Moreover, it is easy to use and interpretable. Future work will focus on developing a dynamic window adjustment mechanism for better anomaly detection on datasets with complex and nonstationary distributions.

Acknowledgments. This research was partially funded by the National Natural Science Foundation of China (NSFC) under grants: 62172112 and 62476063, the National Key Research and Development Program of China under grant: 2022YFE0112200, the Natural Science Foundation of Guangdong Province under grant: 2025A1515011293, and the Guangdong Provincial Key Laboratory of Intellectual Property and Big Data under grant: 2018B030322016.

References

1. Chen, Q., Zhao, M., Ji, Y., Luo, X., Zhang, Y., Zhang, Y.: MGOD: multi-granular outlier detection with clustlier analysis. In: IEEE International Conference on Bioinformatics and Biomedicine, pp. 3105–3110. IEEE (2024)
2. Zhang, C., Chen, X., Tan, Z., Gu, F., Ji, Y., Zhang, Y.: Towards clustering of incomplete mixed-attribute data. Expert Syst. (2025)
3. Chen, J., Ji, Y., Zou, R., Zhang, Y., Cheung, Y.-M.: QGRL: quaternion graph representation learning for heterogeneous feature data clustering. In: ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 297–306 (2024)
4. Zhang, Y., Zhao, M., Chen, Y., Lu, Y., Cheung, Y.-M.: Learning unified distance metric for heterogeneous attribute data clustering. Expert Syst. Appl., 126738 (2025)

5. Feng, S., Zhao, M., Huang, Z., Ji, Y., Zhang, Y., Cheung, Y.-M.: Robust qualitative data clustering via learnable multi-metric space fusion. In: IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 1–5. IEEE (2025)
6. Zhao, M., Zhang, Y., Ji, Y., Lu, Y.: Unsupervised concept drift detection via imbalanced cluster discriminator learning. In: Chinese Conference on Pattern Recognition and Computer Vision, pp. 31–43. Springer (2023)
7. Zhang, Y., Zhang, Y., Lu, Y., Li, M., Chen, X., Cheung, Y.-M.: Asynchronous federated clustering with unknown number of clusters. AAAI Conf. Artif. Intell. **39**(21), 22695--22703 (2025)
8. Costa, B.S.J., Angelov, P.P., Guedes, L.A.: Real-time fault detection using recursive density estimation. J. Control Autom. Electric. Syst. **25**, 428–437 (2014)
9. Mahesh, B.: Machine learning algorithms-a review. Int. J. Sci. Res. **9**(1), 381–386 (2020)
10. Jordan, M.I., Mitchell, T.M.: Machine learning: trends, perspectives, and prospects. Science **349**(6245), 255–260 (2015)
11. Manzoor, E., Lamba, H., Akoglu, L.: xStream: outlier detection in feature-evolving data streams. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1963–1972 (2018)
12. Li, Z., Zhao, Y., Hu, X., Botta, N., Ionescu, C., Chen, G.H.: ECOD: unsupervised outlier detection using empirical cumulative distribution functions. IEEE Trans. Knowl. Data Eng. **35**, 12181–12193 (2022)
13. Almardeny, Y., Boujnah, N., Cleary, F.: A novel outlier detection method for multivariate data. IEEE Trans. Knowl. Data Eng. **34**, 4052–4062 (2020)
14. Basheer, M.Y.I., et al.: Autonomous anomaly detection for streaming data. Knowl.-Based Syst. **284**, 111235 (2024)
15. Da Silva, L.M., et al.: Hardware architecture proposal for TEDA algorithm to data streaming anomaly detection. IEEE Access **9**, 103141–103152 (2021)
16. Xu, H., Pang, G., Wang, Y., Wang, Y.: Deep isolation forest for anomaly detection. IEEE Trans. Knowl. Data Eng. **35**, 12591–12604 (2023)
17. Pevný, T.: LODA: lightweight on-line detector of anomalies. Mach. Learn. **102**, 275–304 (2016)
18. Campello, R.J., Kröger, P., Sander, J., Zimek, A.: Density-based clustering. Wiley Interdisc. Rev. Data Mining Knowl. Discovery **10**(2), e1343 (2020)
19. Bhattacharjee, P., Mitra, P.: A survey of density based clustering algorithms. Front. Comp. Sci. **15**, 1–27 (2021)
20. Peng, M., Wu, Y., Lu, Y., Li, M., Zhang, Y., Cheung, Y.-M.: Weighted density for the win: accurate subspace density clustering. In: IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 1–5. IEEE (2025)
21. Assent, I.: Clustering high dimensional data. Wiley Interdisc. Rev. Data Mining Knowl. Discovery **2**(4), 340–350 (2012)
22. Bijalwan, V., Kumar, V., Kumari, P., Pascual, J.: KNN based machine learning approach for text and document mining. Int. J. Database Theory Appl. **7**(1), 61–70 (2014)
23. Zhang, S., Li, X., Zong, M., Zhu, X., Cheng, D.: Learning K for KNN classification. ACM Trans. Intell. Syst. Technol. **8**(3), 1–19 (2017)
24. Deng, D.: DBSCAN clustering algorithm based on density. In: International Forum on Electrical Engineering and Automation, pp. 949–953 (2020)
25. Zhu, Q., Feng, J., Huang, J.: Natural neighbor: a self-adaptive neighborhood method without parameter K. Pattern Recogn. Lett. **80**, 30–36 (2016)
26. Zou, B., Yang, K., Kui, X., Liu, J., Liao, S., Zhao, W.: Anomaly detection for streaming data based on grid-clustering and Gaussian distribution. Inf. Sci. **638**, 118989 (2023)
27. Aggarwal, C.C.: An Introduction to Outlier Analysis. Springer (2017)

28. Luo, X., Zhang, Y., Ji, Y., Liu, P., Xiao, T.: Efficient topology-driven clustering for imbalanced streaming biomedical data analysis. In: IEEE International Conference on Bioinformatics and Biomedicine, pp. 2262–2267. IEEE (2024)
29. Cheung, Y.-M., Zhang, Y.: Fast and accurate hierarchical clustering based on growing multilayer topology training. IEEE Trans. Neural Netw. Learn. Syst. **30**(3), 876–890 (2018)
30. Zhang, Y., et al.: Learning self-growth maps for fast and accurate imbalanced streaming data clustering. IEEE Trans. Neural Netw. Learn. Syst. (2025)
31. Mariani, G., et al.: A review on the clinical uses of SPECT/CT. Eur. J. Nucl. Med. Mol. Imaging **37**, 1959–1985 (2010)
32. Bloem, B.R., Okun, M.S., Klein, C.: Parkinson's disease. Lancet **397**(10291), 2284–2303 (2021)
33. Keller, F., Muller, E., Bohm, K.: HiCS: high contrast subspaces for density-based outlier ranking. In: IEEE International Conference on Data Engineering, pp. 1037–1048 (2012)
34. Karczmarek, P., Kiersztyn, A., Pedrycz, W., Badurowicz, M., Czerwiński, D., Montusiewicz, J.: K-Medoids clustering and fuzzy sets for isolation forest. In: IEEE International Conference on Fuzzy Systems, pp. 1–8 (2021)
35. Schubert, E., Rousseeuw, P.J.: Faster K-Medoids clustering: improving the PAM, CLARA, and CLARANS algorithms. In: International Conference on Similarity Search and Applications, pp. 171–187 (2019)
36. Chanyaswad, T., Dytso, A., Poor, H.V., Mittal, P.: MVG mechanism: differential privacy under matrix-valued query. In: ACM SIGSAC Conference on Computer and Communications Security, pp. 230–246 (2018)
37. Dong, B., Ju, H., Lu, Y., Shi, Z.: CURE: curvature regularization for missing data recovery. SIAM J. Imag. Sci. **13**(4), 2169–2188 (2020)