Journal of Artificial Intelligence and Data Mining (JAIDM), Vol. 13, No. 3, 2025, 347-358.

Shahrood University of Technology

Journal of Artificial Intelligence and Data Mining (JAIDM)
Journal homepage: http://jad.shahroodut.ac.ir

Research paper

Discrete Rotated Isolation Forest in High Dimensions

Vahideh Monemizadeh and Kourosh Kiani*

Electrical and Computer Engineering Department, Semnan University, Semnan, Iran.

Article Info

Article History:
Received 06 March 2025
Revised 19 May 2025
Accepted 19 June 2025

DOI:10.22044/jadm.2025.15883.2704

Keywords:
Anomaly Detection, Decision Trees, Rotated Isolation Forests, Large-scale High-dimensional Data, Unsupervised Learning, Autoencoder.

**Corresponding author: Kourosh.kiani@semnan.ac.ir (K. Kiani).*

Abstract

Anomaly detection is critical in domains such as cybersecurity, financial risk management, and health monitoring, yet remains challenging due to the complexity of large-scale, high-dimensional, and unlabeled datasets. This paper investigates decision tree-based approaches for their scalability, interpretability, and robustness in such settings. While widely adopted, methods like Isolation Forest (iForest) and Extended Isolation Forest (EIF) often fail to reliably separate anomalies from normal data, occasionally generating undesirable "*ghost clusters*." To address these shortcomings, we previously introduced the Rotated Isolation Forest (RIF) [1], which improved detection accuracy by applying random rotations to the feature space. Expanding on this, we propose the "*Discrete Rotated Isolation Forest (DRIF)",* which integrates an autoencoder for non-linear dimensionality reduction and employs a discrete probability distribution to model random projections more efficiently. The use of an autoencoder improves representation learning by capturing complex structures in the data, while the discrete distribution reduces computational cost and randomness without compromising theoretical soundness. Experimental results on synthetic and real-world datasets show that DRIF consistently outperforms iForest, EIF, and RIF in terms of both ROC-AUC and execution speed. These findings position DRIF as a scalable, efficient, and accurate framework for unsupervised anomaly detection in high-dimensional environments.

1. Introduction

Anomaly detection, the identification of observations that significantly deviate from the norm, is a critical task across diverse fields, including machine learning, data mining, economics, and medicine. Its broad applications span areas such as fraudulent bank transaction detection [2,3], network intrusion detection, cybersecurity of IoT systems [4], machine vision, statistics, and neuroscience.

Categories of Anomaly Detection Techniques

Numerous techniques have been proposed for anomaly detection, generally categorized into three main types based on their reliance on labeled data [5]:

A. Supervised Anomaly Detection

Supervised techniques classify data into "normal" and "abnormal" categories using labeled datasets. They learn the characteristics of both normal and anomalous data to predict the label of new observations. However, this approach faces significant challenges. Firstly, acquiring sufficient labeled data for both normal and, especially, novel anomalies (e.g., in new financial fraud schemes) is often difficult. Secondly, these methods frequently rely on complex deep learning models with millions of parameters, demanding substantial computational resources.

B. Semi-Supervised Anomaly Detection

Semi-supervised methods assume that some portion of the data is labeled, typically focusing on normal instances. They construct a model that fits

the distribution of the labeled normal data. New, unlabeled data points are then evaluated based on their probability of being generated by this model; a low probability suggests an anomaly. Despite their utility, these techniques present three primary challenges: (1) the need for adequate labeled normal data, which may not always be available; (2) the potential for labeled data to be inconsistent with the true normal data distribution; and (3) the inherent complexity and difficulty in explicitly defining models for normal data.

C. Unsupervised Anomaly Detection

Unsupervised techniques are the most general approach as they operate without any prior assumptions about data labels. These methods are particularly valuable when labeled anomaly data is rare or unavailable.

Unsupervised anomaly detection generally involves two main strategies:

Normal Data Modeling: This approach constructs a model or pattern representing normal data. We then identify points that significantly deviate from this model as anomalies. Examples of algorithms that follow this approach are DBScan[6] and K-Nearest Neighbor[7]. However, modelling normal data has certain limitations that we mention two of them next.

- Firstly, the generated model is optimized for normal data, not for anomaly detection, which can lead to high false alarm rates (normal data being misclassified as anomalies) or, conversely, anomalous data being missed (when the normal data model is overly broad). For instance, as shown in Fig. 1, iForest can misclassify points in the tails of normal distributions as anomalies as well as the anomalous points in the middle are classified as normal points.

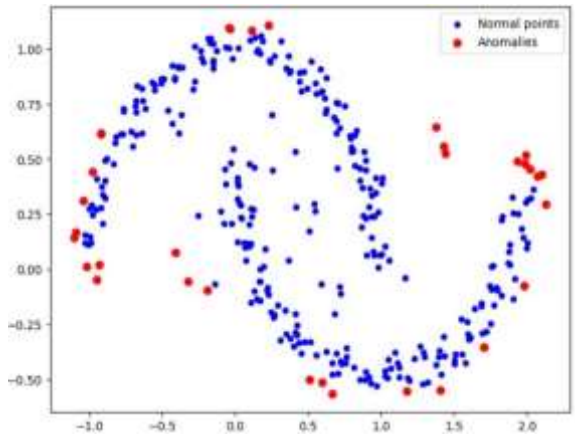


Figure 1. Execution of iForest on the two moons dataset and incorrect identification of the end of the distribution as anomalies.

- Secondly, normal data often do not conform to a simple model, making the generation of an accurate normal data model highly complex.

Isolation of Anomalies: Decision Tree-Based Methods The second primary method for unsupervised anomaly detection focuses on the isolation of anomalous samples rather than finding a model or pattern representing normal data. A popular and highly effective approach within this category are methods based on decision tree learning techniques. These methods, instead of attempting to build an accurate model of normal observations, directly work to separate anomalous points within the dataset from the normal data.

The advantages of decision tree-based anomaly detection methods are significant:

- **Speed:** They are remarkably fast, especially for high-dimensional data.
- **No Labeled Data Required:** These techniques function effectively without the need for labeled data, though incorporating labels can further refine their performance.
- **Interpretability:** Decision trees offer a high degree of interpretability, allowing for insights into why certain observations are flagged as anomalies.

The most prominent method in the class of decision tree algorithms for anomaly detection is the **Isolation Forest (iForest)** algorithm. Its core idea is based on the common fact that anomalous data points are inherently more *"isolated"* and thus require fewer splits to be separated from the rest of the data compared to normal points. iForest was initially introduced by Liu, Ting, and Zhou at ICDM'08 [8] and later published in the highly regarded ACM Transactions on Knowledge Discovery from Data (TKDD) journal in 2012. See reference [9]. Having gained over 8000 citations since 2008, this fundamental paper highlights the important role of iForest in anomaly detection.

Liu et al. [8] conducted a comprehensive empirical analysis of the Isolation Forest (iForest) algorithm. Their experiments demonstrated that iForest is highly efficient in both computation time and memory usage. This efficiency stems from its use of random sampling and tree-based partitioning, which allow the algorithm to operate with linear time complexity and minimal memory requirements. They also found that iForest performs well on high-dimensional datasets, making it a strong choice for practical anomaly detection tasks.

While Isolation Forest (iForest) is a powerful and widely used anomaly detection technique, it has known limitations. Hariri et al. [10] identified some

of these drawbacks—most importantly the *"ghost clusters" phenomenon*—and proposed an enhanced variant called the Extended Isolation Forest (EIF) to address them. However, in our recent work [1], we showed that EIF also suffers from ghost clusters, albeit in a different form. Specifically, while ghost clusters in iForest tend to appear outside the regions of normal data, EIF tends to exhibit them in the areas between normal clusters.

Limitations of IForest and EIF

The main issue with iForest is that iForest has inherent limitations due to its axis-aligned partitioning strategy. Specifically, this bias arises from the way isolation trees' branch—using hyperplanes that are always perpendicular to one of the feature axes. As demonstrated by Hariri et al. [10], this constraint can lead to unintended consequences in high-dimensional spaces which is known as the formation of "ghost clusters": regions where anomalous points are assigned low anomaly scores, falsely categorizing them as normal.

This issue typically emerges at the intersections of axis-aligned hyperplanes, as indicated by the black circles in Fig. 2, leading to inaccurate scoring and potentially erroneous predictions.

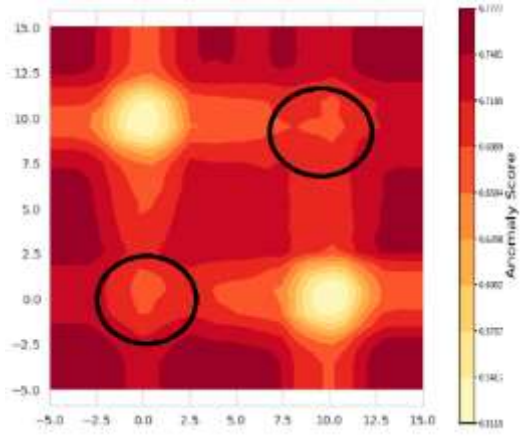


Figure 2. Anomaly score map of iForest and the presence of two ghost clusters, indicated by black circles.

To address this limitation, Hariri et al. proposed the Extended Isolation Forest (EIF) [10], which modifies the tree-splitting mechanism to reduce this directional bias. Instead of selecting a random axis-aligned split, EIF selects a random point p and a random direction vector n, and uses them to define a hyperplane that partitions the space in a more flexible and general way. This approach enables splits in arbitrary orientations, not just those aligned with coordinate axes.

As illustrated in Fig. 3, EIF successfully eliminates some of the ghost clusters observed in iForest. However, a new ghost cluster appears near the center of the figure, where anomaly points are again misclassified as normal. This suggests that while EIF improves upon iForest's partitioning bias, it does not fully resolve the ghost cluster phenomenon.

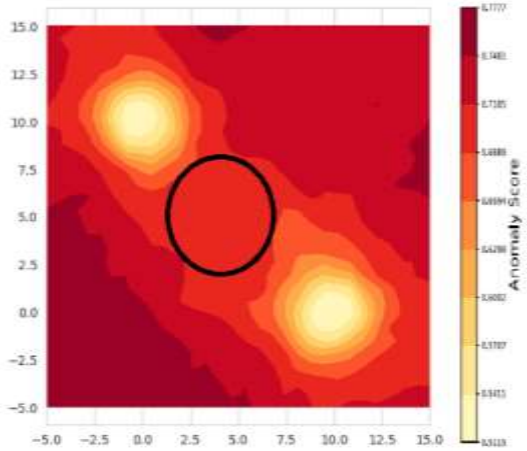


Figure 3. Anomaly score map of EIF showing a ghost cluster at the center, indicated by the black circle.

Our Contribution: To address these issues, we recently introduced the Rotated Isolation Forest (RIF) [1], which improves anomaly detection by applying random rotations to the feature space.

Our proposed algorithm builds upon the Rotated Isolation Forest (RIF) framework. However, we introduce several key innovations aimed at both improving detection performance and significantly reducing computation time. First, we use *autoencoders* to perform dimensionality reduction on the input data. This not only accelerates the anomaly detection process by reducing the number of dimensions but also helps identify *latent* features that capture the essential structure of normal data. By focusing on these informative dimensions, our model becomes more effective at distinguishing anomalies from normal patterns.

We call our new anomaly detection method the Discrete Rotated Isolation Forest (DRIF). One of the core enhancements in DRIF is the replacement of Gaussian-based random matrix generation—used in RIF—with discrete probability distributions. While traditional techniques such as Principal Component Analysis (PCA) [11] and the Johnson-Lindenstrauss (JL) [12] Lemma often rely on Gaussian distributions, more recent and efficient variants of these methods employ discrete distributions for faster and more scalable computation. Discrete distributions not only reduce computational overhead but also produce sparser transformations, which further accelerate dimensionality reduction and improve overall algorithm performance.

Experimental results on both synthetic and real-world datasets show that DRIF outperforms

iForest, EIF, and RIF—achieving higher Receiver Operating Characteristic–Area Under the Curve (ROC-AUC) scores and at the same time, DRIF has significantly faster execution times than these techniques.

2. Further Related Work

Anomaly detection algorithms generally fall into several categories:

A. Classification-Based Methods

These methods learn parameters from training data to build classification models for anomaly detection. Examples include neural networks [13], Bayesian networks [14], support vector machines [15], and rule-based models [16].

While these models can offer good and efficient detection performance, their effectiveness is often limited by the availability of high-quality, relevant training data.

B. Distance-Based Methods

Distance-based methods identify anomalies based on the proximity of data points to each other. A key advantage of these techniques is their unsupervised nature, eliminating the need for labeled data. This category can be further divided:

Nearest Neighbor Techniques: These techniques quantify the (ab)normality of a point using its distance to the $k-th$ nearest neighbor or other local neighborhood metrics. For instance, the Local Outlier Factor (LOF) uses k-nearest neighbors to compute a relative density value for a point (similar to DBSCAN concepts) to address density variations across different data clusters.

Clustering: These methods group data into clusters, with points that do not belong to any cluster, or are far from any cluster centroid, being identified as anomalies. Popular clustering algorithms used here include k-means [7] and DBSCAN [6], both of which rely on distance criteria.

C. Statistics-Based Methods

These techniques employ statistical distributions, such as Gaussian models [17] or regression [18], to identify anomalies. Observations that deviate significantly from the learned statistical model are flagged as anomalous.

D. Isolation-Based Methods

Isolation-based anomaly detection algorithms work by directly isolating anomalies from normal data, often through a series of "cuts" or partitions. Experimental evidence, including studies by Liu et al. [8], suggests that isolation-based algorithms, particularly those leveraging randomization techniques, often outperform other types of anomaly detection, such as distance-based [19,20,21,22,23] and density-based methods [24,25,26].

Forest (iForest) [8] is the seminal algorithm in this class. Recently, several variations of iForest have been proposed, including Extended Isolation Forest (EIF) [10] and Robust Random Pruning Forest (RRCF) [27].

3. Proposed Model

In the following sections, we first provide a brief overview of the iForest and RIF algorithms to establish the necessary background. We then introduce autoencoders and explain how they contribute to both dimensionality reduction and feature extraction in DRIF. Next, we discuss the role of discrete probability distributions in our approach and why they offer advantages over Gaussian-based methods.

Finally, we present the complete DRIF algorithm, where we explain the details of its construction and integration of the proposed techniques.

3.1 Rotated Isolation Forest (RIF)

Isolation Forest (iForest) is a popular algorithm for unsupervised anomaly detection that isolates anomalies instead of profiling normal points. It operates by constructing a set of binary trees, known as isolation trees (iTrees), where each split randomly selects a feature and a split value. Since anomalies tend to be rare and different, they are easier to isolate and usually appear closer to the root of the tree.

The average path length from the root to a point across all trees is used as an anomaly score, with shorter paths indicating higher anomaly likelihood. The iForest is computationally efficient and works well on low- to moderate-dimensional data, but its effectiveness can degrade in high-dimensional settings where axis-aligned splits fail to capture complex data structure.

The Rotated Isolation Forest (RIF) extends iForest by addressing this limitation through random feature space rotations. In high-dimensional spaces, data often lies along correlated or skewed directions that axis-aligned splits in iForest cannot effectively isolate. RIF introduces a preprocessing step that rotates the data using randomly generated orthogonal matrices before building each isolation tree. By rotating the feature space, RIF enhances the diversity of tree structures and improves the algorithm's ability to distinguish anomalies that may otherwise be aligned with dominant directions in the original feature space.

The generation of these random rotation matrices is a key part of RIF. It involves two main steps: first, a random square matrix is created by drawing each element independently from a standard normal distribution. Then, this matrix is decomposed using a mathematical technique called *QR* decomposition, which yields an orthogonal matrix that serves as a rotation matrix. This orthogonal transformation preserves the geometric structure of the data—maintaining distances and angles between points—while changing the orientation of the feature space. By applying a different random rotation before building each tree, RIF produces a more expressive ensemble that can better isolate subtle or hidden anomalies.

During inference, a test point is rotated using the same set of matrices used during training and passed through each corresponding isolation tree. The anomaly scores from each tree are then aggregated to compute the final anomaly score for the point. This consistent use of rotation during both training and scoring ensures that the benefits of the transformed space are fully realized. Overall, RIF retains the efficiency and scalability of iForest while significantly enhancing its ability to detect anomalies in complex, high-dimensional datasets.

3.2 Dimensionality Reduction Using Autoencoders

An autoencoder is a deep neural network commonly used for unsupervised dimensionality reduction, especially in high-dimensional datasets. It learns an efficient compression of the input data by training to reconstruct the original input from a compressed representation. The network consists of two parts: an encoder, which maps the input data into a lower-dimensional latent space, and a decoder, which reconstructs the original data from this compressed form.

In our approach, we employ autoencoders to reduce the dimensionality of datasets, as illustrated in Table 3 and will be explained later, which includes the size, number of dimensions, and the distribution of normal and anomalous samples. This dimensionality reduction accelerates anomaly detection and helps extract meaningful features that better characterize normal data behavior.

Setup autoencoder hyperparameters: The hyperparameters for the autoencoder were selected based on standard practices commonly used in unsupervised feature learning, especially for anomaly detection. Our goal was to balance expressive power, training stability, and computational efficiency across various datasets, especially those with high dimensionality. Below are the specific choices and justifications:

- *Encoding dimension = 10:* We selected a 10-dimensional bottleneck layer as a simple yet effective choice to compress the original high-dimensional data into a lower-dimensional representation. This dimensionality was chosen empirically, and we found it to preserve important structure in the data while improving downstream anomaly detection.
- *Epochs = 20 and Batch size = 32:* These are typical default values for training autoencoders on medium-sized datasets. They provide sufficient convergence while avoiding overfitting.
- *Activation = 'ReLU':* ReLU activation is widely used in deep learning due to its simplicity and ability to alleviate the vanishing gradient problem.
- *Optimizer = 'Adam':* The Adam optimizer is chosen for its adaptive learning rate capabilities, which generally lead to faster and more stable convergence.
- *Loss = 'Mean Squared Error':* Since our goal is to reconstruct the input as closely as possible, MSE is an appropriate choice for continuous-valued inputs.
- *Architecture:* We used a single dense layer for the encoder and decoder, keeping the architecture shallow to reduce training complexity and avoid overfitting given our limited number of epochs.

While we acknowledge that more complex architectures and hyperparameter optimization could further improve performance, our focus in this work was to evaluate the core contribution of DRIF.

3.3 Discrete Probability Distributions for Rotation Matrices

A key innovation in our model is the use of discrete probability distributions to generate random rotation matrices, instead of traditional continuous Gaussian distributions. For the discrete probability distribution used to generate the rotation matrix, our settings are based on the result that Achlioptas [2] obtained for Johnson-Lindenstrauss Lemma [12]. In particular, he showed that random projections satisfying the Johnson–Lindenstrauss Lemma (JL-Lemma) [12] can be constructed using a simple sparse distribution where in the random matrix that we generate each matrix entry is set to $+1$ with probability 1/6, is set to -1 with probability 1/6, and to 0 with probability 2/3. The configuration that he devised for JL-Lemma ensures two key benefits:

1. **Variance preservation**: The sparse random matrix still preserves pairwise distances between data points with high probability.
2. **Sparsity**: With 2/3 of the entries being zero, the resulting matrix of dataset is computationally efficient for matrix multiplications, which is particularly valuable for high-dimensional data.

In light of the theoretical support established in his work, we adopt the discrete probability distribution he introduced." Indeed, to create a random matrix A of size $d \times d$ (where d is the reduced dimension after autoencoding), each element $a[i,j]$ is independently sampled according to the following discrete distribution:

- With probability $\frac{1}{2x}$, the element is set to -1.
- With probability $\frac{1}{2x}$, the element is set to +1.
- With probability $1-\frac{1}{x}$, the element is set to 0.

For example, when $x=6$, each element has a $\frac{1}{12}$ chance to be –1, a $\frac{1}{12}$ chance to be +1, and a $\frac{5}{6}$ chance to be 0 [28]. This creates a sparse matrix where approximately 83% of the elements are zero, which significantly speeds up matrix multiplications.

After generating this sparse random matrix A, we perform QR decomposition to factorize it into an orthogonal matrix Q and an upper triangular matrix R. The orthogonal matrix Q is then used as the random rotation matrix in our algorithm. Because Q is derived from a sparse random matrix, it retains the orthogonality properties necessary to preserve distances and angles, while enabling faster computations compared to dense Gaussian-based rotations.

3.4 Discrete Rotated Isolation Forest (DRIF)

The proposed Discrete Rotated Isolation Forest (DRIF) algorithm combines the advantages of autoencoder-based dimensionality reduction with efficient discrete random rotations. The algorithm consists of two main stages:

1. **Dimensionality Reduction:** The input data is compressed into a lower-dimensional space using an autoencoder. This step extracts important features that characterize normal data, which facilitates faster and more accurate anomaly detection.
2. **Discrete Rotated Isolation Forest:** Using the reduced-dimension data, an isolation forest is constructed where each tree is trained on data rotated by an orthogonal matrix generated from the discrete probability distribution described above. This approach improves computational efficiency and enhances anomaly detection by introducing diverse data orientations.

Figure 4 illustrates the DRIF architecture, showing how the autoencoder reduces dimensions before the discrete rotation and isolation forest stages. The combination of sparse rotations and dimensionality reduction allows DRIF to scale efficiently to high-dimensional datasets without sacrificing detection performance.

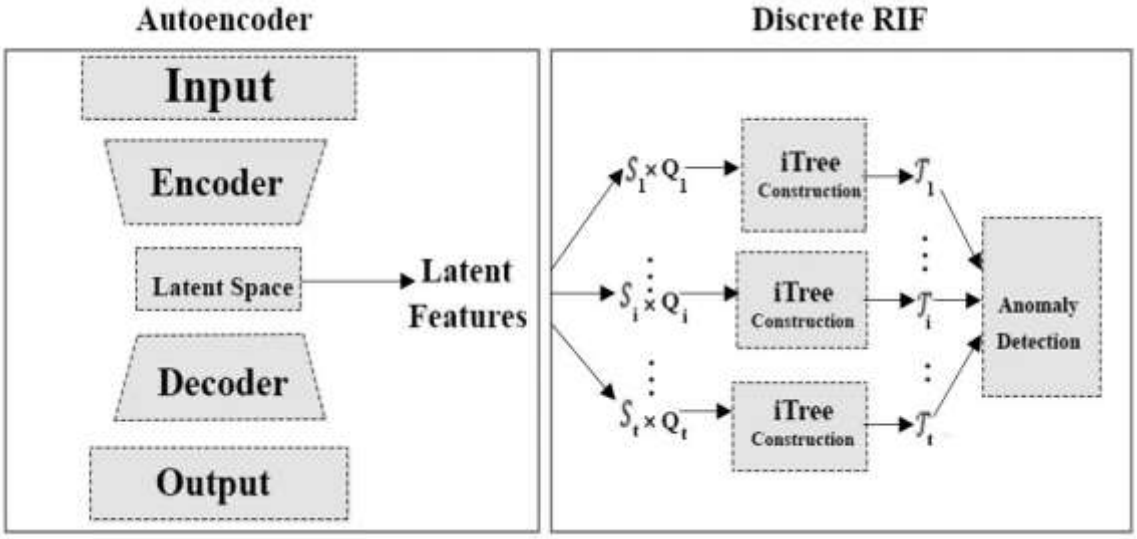


Figure 4. Diagram of the Proposed Model (DRIF).

4. Experimental Results

In this section, we present the experiments that we have conducted to evaluate the performance and runtime of DRIF in comparison with iForest, EIF, and RIF. We begin by describing the datasets used in our study, which include both synthetic as well as real-world data, spanning a range of dimensionalities from low to high. We then discuss the results obtained from the synthetic datasets.

A key advantage of using synthetic data is that, since we generate the normal instances and inject anomalies in two dimensions, it becomes easy to visualize the separation between normal data and anomalies. This also allows us to precisely measure and analyze the performance of each algorithm. Finally, we evaluate the performance and efficiency of DRIF, iForest, EIF, and RIF on real-world datasets, many of which are high-dimensional. This enables us to assess and compare the scalability and speed of these algorithms in more complex scenarios.

4.1 Datasets

We categorize the datasets that we have used in our experiments into two groups: synthetic datasets and real-world datasets. Below, we introduce and describe each category in more detail. All experiments involving synthetic and real datasets were executed on the Google Colab platform.

Synthetic Datasets: The synthetic datasets were generated using the scikit-learn[1] library and include well-known dataset types such as *blobs*, *Aniso*, and *Sine* (Sinusoid). Each dataset comprises 10,000 normal data points and 1,000 anomalous data points, randomly mixed. These synthetic datasets are two-dimensional, which makes them ideal for visualization and detailed analysis of how each algorithm identifies normal and anomalous patterns. The characteristics of the synthetic datasets, including their structure and data distribution, are summarized in Table 1.

Table 1. Features of the synthetic datasets.

Datasets	Size	Dimensions	Anomaly Numbers
blobs	11000	2	1000
Aniso	11000	2	1000
Sinuside	11000	2	1000

Real-World Datasets: We used 13 real-world datasets to evaluate the algorithms under more practical and diverse conditions. These datasets include: Http [29], Mammography [29], Shuttle [29], Pima [29], Cardio [29], Satellite [29], Smtp [29], Oil-Spill [30], SpamBase [31], Backdoor [32], Scene [33], Census [32], and Madelon [34].

We further group these datasets based on their dimensionality: The first seven datasets (Http through Smtp) have dimensions ranging from 3 to 38, which we classify as low to moderate dimensional. The remaining six datasets (Oil-Spill through Madelon) are high-dimensional, with feature dimensions ranging from 50 up to 501.

We conducted a comprehensive set of experiments using these real-world datasets to evaluate the performance and runtime of four algorithms: iForest, EIF, RIF [1], and DRIF. These experiments help us understand how the algorithms behave under both low and high-dimensional scenarios.

Detailed information about each real-world dataset is presented in Tables 2 and 3. These tables includes: the dataset name (first column), the number of instances (second column), the dimensionality of the data (third column), as well as the number of normal and anomalous samples, shown in the fifth and sixth columns respectively.

Table 2. Features of the real datasets.

Data sets	Size	Dimensions	Anomaly Numbers	Anomaly Label	Normal Label
Http	567497	3	2213	1	0
Mammography	11183	6	259	1	-1
Shuttle	57990	9	3501	1	0
Pima	1832	21	641	1	0
Cardio	1831	21	190	1	0
Satellite	6435	36	2059	Anomaly	Normal
Smtp	96554	38	1183	1	0

This setup allows for a thorough comparison of algorithm performance across a diverse set of conditions and data complexities.

[1] https://scikit-learn.org/stable/api/sklearn.datasets.html

Table 3. Features of the high dimensional real datasets.

Dataset	Size	Dimension	Normal Label	Anomaly Label	Contamination
Oil-Spill	937	50	-1	1	42 (0.04)
SpamBase	4601	58	0	1	1814 (0.39)
backdoor	95329	196	0	1	2330 (0.02)
scene	2407	300	1	0	432 (0.18)
census	299285	500	0	1	18569 (0.06)
madelon	2600	501	0	1	1301 (0.50)

4.2. Results on Synthetic Datasets

As outlined previously, we first benchmarked the performance of DRIF against iForest, EIF, and RIF using a set of synthetic datasets. These datasets include One Blob, Two Blobs, Aniso, and Sinusoid, all generated using the scikit-learn library. Each dataset contains 10,000 normal data points and 1,000 injected anomalies. These two-dimensional datasets allow for straightforward visualization of anomaly detection behavior, making them well-suited for qualitative analysis.

The results, illustrated through heatmaps in Figures 5 to 8, highlight the superiority of DRIF in handling various synthetic data distributions. Interestingly, RIF and DRIF produced almost identical detection patterns, indicating that our model closely matches or surpasses the performance of RIF, which is itself an improvement over traditional isolation-based methods.

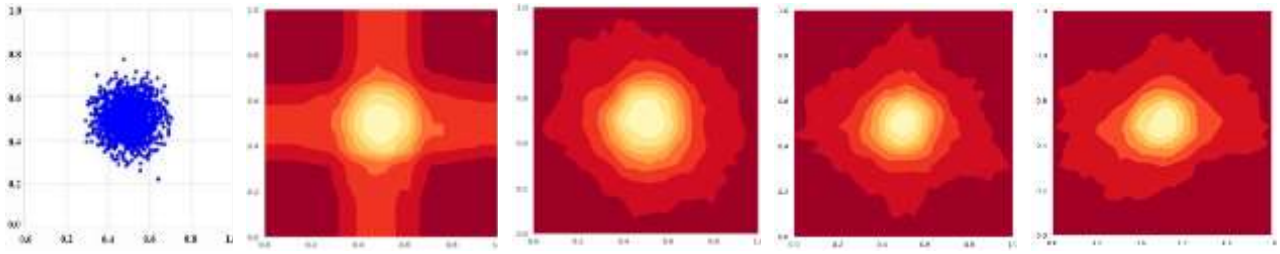
One Blob Dataset (Figure 5): This dataset contains a single Gaussien cluster of normal data points with anomalies scattered around it. The iForest algorithm exhibits a well-documented artifact—formation of vertical and horizontal decision boundaries due to axis-parallel splits—which results in ghost clusters appearing along the x and y axes. These ghost clusters incorrectly mark regions of normal data as anomalous. While EIF reduces this effect through the use of random hyperplanes, it still misclassifies boundary points and shows less accurate anomaly localization than DRIF. DRIF demonstrates more coherent and compact anomaly regions, with no evidence of ghost clusters.

Two Blobs Dataset (Figure 6): In this dataset, two separate clusters of normal points are placed symmetrically, with anomalies surrounding them. Here, iForest again produces ghost clusters, particularly in the northeast and southwest quadrants, far from any true normal data. EIF partially mitigates these artifacts but introduces a new ghost cluster in the center of the plot, where no normal data exists. In contrast, DRIF accurately preserves the separation between the two normal clusters and avoids creating any artificial anomaly zones, demonstrating its improved spatial representation and anomaly boundary modeling.

Aniso Dataset (Figure 7): This dataset includes a single cluster of normal points with an anisotropic (skewed) covariance structure, introducing directional variance. Both iForest and EIF struggle to adapt to the elongated shape of the data, once again forming ghost clusters and misclassifying large areas as anomalies. DRIF, by contrast, effectively adapts to the skewed distribution and correctly identifies the boundaries of the anomaly regions without introducing spurious detection zones.

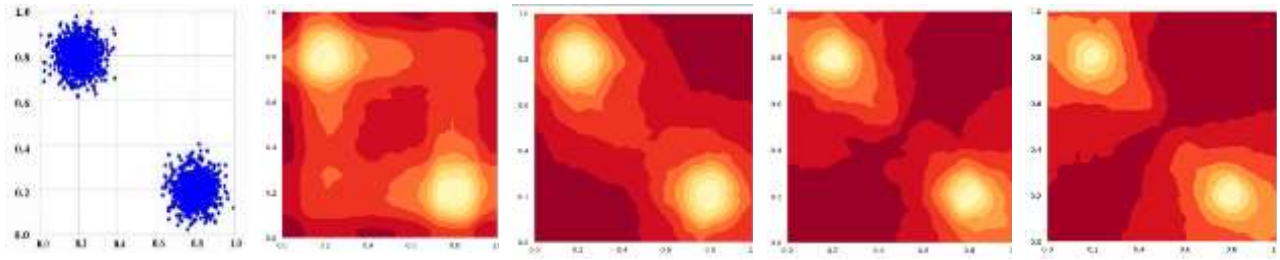
Sinusoid Dataset (Figure 8): The final synthetic dataset features a sinusoidal wave pattern, where normal data follows a nonlinear curve. This complex structure poses a challenge for algorithms that rely on axis-aligned or linear partitioning. Both iForest and EIF fail to trace the underlying structure of the data, resulting in ghost clusters and poor detection accuracy. DRIF stands out in this scenario, successfully identifying the shape of the sinusoid and accurately separating normal and anomalous regions.

This section demonstrates that DRIF consistently outperforms iForest and EIF across various synthetic settings by avoiding ghost clusters and providing more precise anomaly boundaries. The visual comparisons in Figures 5 through 8 support this conclusion and highlight DRIF's robustness to different data distributions.



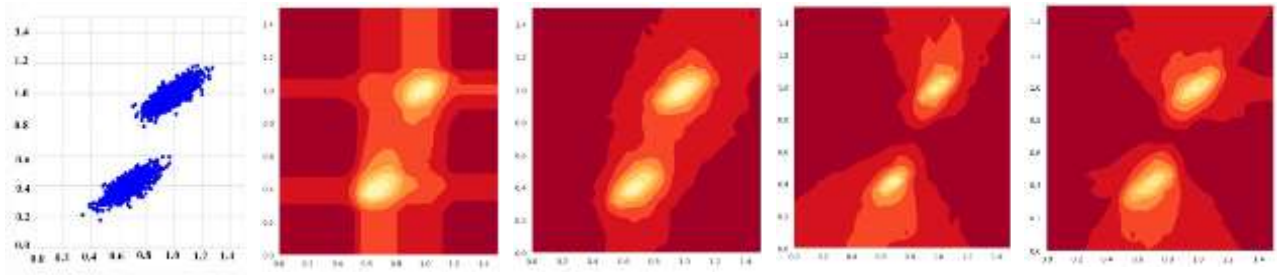
One Gaussian **iForest** **EIF** **RIF** **Our Model**

Figure 5. Anomaly score map of iForest, EIF, RIF and our model on one Gaussian distribution.



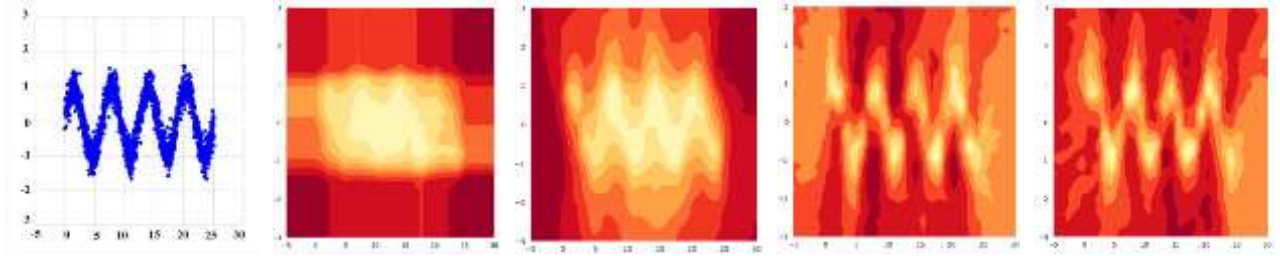
Two Gaussian **iForest** **EIF** **RIF** **Our Model**

Figure 6. Anomaly score map of iForest, EIF, RIF and our model on two Gaussian distribution.



Aniso **iForest** **EIF** **RIF** **Our Model**

Figure 7. Anomaly score map of iForest, EIF, RIF and our model on Aniso distribution.



Sinuside **iForest** **EIF** **RIF** **Our Model**

Figure 8. Anomaly score map of iForest, EIF, RIF and our model on Sinuside distribution.

4.3. Results on real datasets

An anomaly detection algorithm operates similarly to a binary classifier [35], as it attempts to distinguish between normal and anomalous instances within a dataset. Let us consider an arbitrary dataset D that contains n total entries, of which a subset is labeled as anomalies based on ground truth. The remaining entries represent normal data. The contamination ratio, denoted as v, is defined as the proportion of anomalies to the total number of entries. This value reflects the level of class imbalance typically found in anomaly detection tasks.

To evaluate the effectiveness of anomaly detection algorithms, we use the "*Area Under the Receiver Operating Characteristic Curve (AUC-ROC)*" [36]. AUC is a widely used metric in binary classification [35] that quantifies the model's ability to separate positive (anomalous) and negative (normal) instances across all possible decision thresholds. Specifically, the AUC score represents the probability that a randomly selected anomalous instance is ranked higher (i.e., assigned a higher anomaly score) than a randomly selected normal instance. An AUC score of 1.0 indicates perfect classification, while a score of 0.5 reflects random guessing.

AUC is particularly advantageous over metrics such as accuracy, especially for imbalanced datasets, as it considers both true positive and false positive rates across thresholds. This makes AUC a robust and threshold-independent evaluation metric that provides a comprehensive view of an algorithm's predictive performance.

Our implementation consists of two main components:

A. Comparison of Discrete Distribution in DRIF vs. Continuous Distribution in RIF. In the first set of experiments, we compare the performance of the Random Isolation Forest (RIF) algorithm proposed in [1], which generates random rotation matrices using a continuous normal distribution, with our proposed Discrete Random Isolation Forest (DRIF), which uses a discrete probability distribution. As discussed in Section 3.3, we hypothesize that discrete sampling leads to faster execution while maintaining high detection accuracy. Table 4 presents the comparative results in terms of AUC scores across multiple datasets. The DRIF model consistently outperforms iForest, EIF, and the original RIF in most cases. For instance, in the Cardio dataset:

- iForest achieves an average AUC of 0.84,
- EIF scores 0.84,
- RIF scores 0.89,
- DRIF achieves the highest score of 0.90.

These results confirm that DRIF not only maintains high anomaly detection performance but also offers significant improvements in computational efficiency. Runtime comparisons in Table 4 further validate our claim from Section 3.3 that using a discrete distribution results in faster execution times.

B. Dimensionality Reduction with Autoencoder and DRIF. In the second part of our evaluation, we integrate autoencoders for dimensionality reduction prior to applying the anomaly detection algorithms. This approach is particularly useful for high-dimensional datasets, as it compresses the data while preserving its essential structure.

We apply the iForest, EIF, RIF, and DRIF algorithms on the encoded representations, with DRIF utilizing the discrete probability distribution for generating rotation matrices. The results of these experiments are summarized in Table 5. For each dataset and algorithm combination, we repeated the experiments five times and report the average AUC score to ensure robustness and reliability.

Execution time was also recorded to evaluate computational efficiency. As shown in Table 5, DRIF significantly outperforms iForest, EIF, and RIF in terms of speed across all six high-dimensional datasets. These results demonstrate the combined benefits of autoencoder-based dimensionality reduction and discrete rotation matrix generation, leading to improved performance and reduced execution time.

Table 4. The results of real datasets for iForest, EIF, RIF, and Discrete RIF Distribution. In this table, Cont" is abbreviation for "Contamination". Bold numbers correspond to the best result.

Dataset		iForest		EIF		RIF		Discrete RIF Distribution		
	Dim	AUC ROC	Time (s)	AUC ROC	Time (s)	AUC ROC	Time (s)	AUC ROC	Time (s)	Cont
Http	3	0.89	**1379**	0.91	1384	**0.98**	1445	0.97	1412	0.05
Mammography	6	0.79	30.1	0.79	30.3	0.80	30.3	**0.82**	**30.0**	0.23
Shuttle	9	0.97	158	0.97	160	0.98	139	**0.99**	**137**	0.08
Cardio	21	0.84	4.7	0.84	4.7	0.89	4.8	**0.90**	**4.6**	0.23
Pima	21	0.64	2.3	0.63	2.3	0.65	2.5	**0.66**	**2.2**	0.40
Satellite	36	0.70	21	0.71	21	0.73	20	**0.74**	**17**	0.23
Smtp	38	0.82	**242**	0.81	247	0.82	259	**0.83**	253	0.05
Oil-Spill	50	0.64	4	0.68	3	0.76	3	**0.77**	**2.6**	0.25
SpamBase	58	0.56	12	0.57	12	**0.65**	13	0.59	**11.9**	0.35
Backdoor	196	0.74	329	0.75	326	0.76	325	**0.77**	**270**	0.45
scene	300	0.54	**13**	0.55	13	**0.61**	29	**0.61**	18	0.40
census	500	0.58	1370	0.56	1323	0.58	948	**0.61**	**860**	0.45
madelon	501	0.49	20	0.50	**18**	0.51	52	**0.52**	33	0.10

Table 5. The results of real datasets for iForest, EIF, RIF, and DRIF. In this table, "Red-Dim", and "Cont" are abbreviations for "Reduced Dimensions", and "Contamination", respectively. Bold numbers correspond to the best result.

Dataset	Dim	iForest		EIF		RIF		DRIF			
		AUC ROC	Time (s)	AUC ROC	Time (s)	AUC ROC	Time (s)	AUC ROC	Time (s)	Red-Dim	Cont
Oil-Spill	50	0.64	4	0.68	3	0.76	3	**0.77**	**3**	10	0.25
SpamBase	58	0.56	12	0.57	12	0.65	13	**0.68**	**9**	10	0.35
Backdoor	196	0.74	329	0.75	326	0.76	325	**0.77**	**151**	5	0.45
scene	300	0.54	13	0.55	13	0.61	29	**0.62**	**8**	10	0.40
census	500	0.58	1370	0.56	1323	0.58	948	**0.64**	**759**	20	0.45
madelon	501	0.49	20	0.50	18	0.51	52	**0.52**	**8**	10	0.10

5. Conclusion

This paper introduced DRIF, a novel extension of the Random Isolation Forest (RIF) [37] framework for unsupervised anomaly detection, particularly effective in high-dimensional and complex datasets. The proposed model addresses several key limitations observed in existing algorithms, such as iForest and EIF, including the formation of ghost clusters and reduced detection accuracy in the presence of non-linear or skewed data distributions.

DRIF enhances the anomaly detection pipeline in two major ways. First, it employs a discrete probability distribution to generate random rotation matrices, replacing the continuous Gaussian sampling used in RIF. This substitution significantly reduces computational overhead while maintaining or improving detection performance. Second, DRIF incorporates dimensionality reduction via autoencoders, enabling it to handle high-dimensional datasets more efficiently without compromising the integrity of the data's underlying structure.

Extensive experiments on both synthetic and real-world datasets validate the effectiveness of DRIF. On synthetic datasets, DRIF avoids ghost clusters and consistently provides more accurate anomaly boundaries. On real-world datasets, DRIF outperforms iForest, EIF, and RIF in terms of ROC-AUC scores and demonstrates superior execution speed, making it scalable and robust across a wide range of data complexities. Overall, DRIF offers a practical, high-performance solution for real-world anomaly detection tasks. Its improved accuracy, efficiency, and adaptability make it a compelling alternative to existing isolation-based methods in diverse application domains.

Future Work. There are several promising directions for future research. First, exploring sparser and structured random rotation matrices could further improve the efficiency of the approach. Second, incorporating *k*-wise independent random variables may help reduce the amount of randomness required while maintaining theoretical guarantees. Third, extending DRIF to semi-supervised settings, where only partial labels are available, could broaden its applicability. Finally, investigating more advanced autoencoder architectures along with systematic hyperparameter tuning may lead to further improvements in representation learning.

References

[1] V. Monemizadeh, K. Kiani. “Detecting anomalies using rotated isolation forest.” *Data Min Knowl Disc* 39, 24 (2025).

[2] Chugh, Bharti, Nitin Malik, Deepak Gupta, and Badr S. Alkahtani. “A probabilistic approach driven credit card anomaly detection with CBLOF and isolation forest models.” *Alexandria Engineering Journal* 114 (2025).

[3] Buchdadi, A. Dharmawan, and A. Salamh Mujali Al-Rawahna. “Anomaly Detection in Open Metaverse Blockchain Transactions Using Isolation Forest and Autoencoder Neural Networks.” *International Journal Research on Metaverse* 2, no. 1 (2025).

[4] R. Morshedi, and S. Mojtaba Matinkhah. “Anomaly Detection in IoT Traffic in the Presence of Gaussian Noise Using Deep Neural Networks.” *Journal of AI and Data Mining* (2025).

[5] .VChandola, A. Banerjee, V. Kumar. “Anomaly detection: A survey.” *ACM Computing Surveys.* 41 (3): 1–58. (2009).

[6] M. Çelik, F, Dadaşer-Çelik, and A, Şakir Dokuz. “Anomaly detection in temperature data using dbscan algorithm.” *In 2011 international symposium on innovations in intelligent systems and applications*, pages 91–95. IEEE, (2011).

[7] G. Münz, Sa Li, and G. Carle. “Traffic anomaly detection using k-means clustering.” *In GI/ITG Workshop MMBnet*, volume 7, page 9, (2007).

[8] F. Tony Liu, K. Ming Ting, and Z. Zhou. “Isolation forest.” *In 2008 eighth ieee international conference on data mining*, pages 413–422. IEEE, (2008).

[9] Liu, F. Tony, K. Ming Ting, and Z. Zhou. “Isolation-based anomaly detection.” *ACM Transactions on Knowledge Discovery from Data (TKDD)* 6, no. 1 (2012).

[10] S. Hariri, M. Carrasco Kind, and R. J. Brunner. “Extended isolation forest.” *IEEE transactions on knowledge and data engineering* 33, no. 4 (2019).

[11] A. Maćkiewicz, and W. Ratajczak. “Principal components analysis (PCA).” *Computers & Geosciences* 19, no. 3 (1993).

[12] B. W. Johnson, and J. Lindenstrauss. "Extensions of Lipschitz mappings into a Hilbert space." *Contemporary mathematics* 26, no. 189-206 (1984).

[13] R. Chalapathy, A. Krishna Menon, and S. Chawla. “Anomaly detection using one-class neural networks.” *arXiv preprint arXiv*:1802.06360, (2018).

[14] S. Mascaro, A. E Nicholso, and K. B Korb. “Anomaly detection in vessel tracks using bayesian networks.” *International Journal of Approximate Reasoning*, 55(1):84–98, (2014).

[15] K. Li, H. Huang, S. Tian, and W. Xu. “Improving one-class svm for anomaly detection.” *In Proceedings of the 2003 international conference on machine learning and cybernetics (IEEE Cat. No. 03EX693)*, volume 5, pages 3077–3081. IEEE, (2003).

[16] N. Duffield, P. Haffner, B. Krishnamurthy, and H. Ringberg. “Rule-based anomaly detection on ip flows.” *In IEEE INFOCOM* 2009, pages 424–432. IEEE, (2009).

[17] R. Laxhammar. “Anomaly detection for sea surveillance.” *In 2008 11th international conference on information fusion*, pages 1–8. IEEE, (2008).

[18] O. Salem, A. Guerassimov, A. Mehaoua, A. Marcus, and B. Furht. “Anomaly detection in medical wireless sensor networks using svm and linear regression models.” *International Journal of E-Health and Medical Communications (IJEHMC)*, 5(1):20–45, (2014).

[19] M. M. Breunig, H.-Peter Kriegel, R. T. Ng, and J. Sander. “Lof: Identifying density-based local outliers.” *SIGMOD* , page 93–104, New York, NY, USA, (2000).

[20] J. Tang, Z. Chen, A. Wai-Chee Fu, and D. W Cheung. “Enhancing effectiveness of outlier detections for low density patterns.” *In Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 535–548. Springer, (2002).

[21] S. Papadimitriou, H. Kitagawa, P. B Gibbons, and C. Faloutsos. “Loci: Fast outlier detection using the local correlation integral.” *In Proceedings 19th international conference on data engineering* (Cat. No. 03CH37405), pages 315–326. IEEE, (2003).

[22] W. Jin, A. KH Tung, J. Han, and W. Wang. “Ranking outliers using symmetric neighborhood relationship.” *In Pacific-Asia conference on knowledge discovery and data mining*, pages 577–593. Springer, (2006).

[23] H.-Peter Kriegel, P. Kröger, E. Schubert, and A. Zimek. “Loop: local outlier probabilities.” *In Proceedings of the 18th ACM conference on Information and knowledge management*, pages 1649–1652, (2009).

[24] V. Chandola, A. Banerjee, and Kumar. “Anomaly detection: A survey.” *In Computing Surveys* 41, 3, pages 1–58, (2009).

[25] P. N. Tan, M. Steinbach, and V. Kumar. “Introduction to data mining.” *Addison-Wesley*, (2005).

[26] M. Ester, H.-Peter Kriegel, J. Sander, X. Xu, et al. “A density-based algorithm for discovering clusters in large spatial databases with noise.” *In Kdd*, volume 96, pages 226–231, (1996).

[27] S. Guha, N. Mishra, G. Roy, and O. Schrijvers. “Robust random cut forest-based anomaly detection on streams.” *In International conference on machine learning*, pages 2712–2721. PMLR, (2016).

[28] Achlioptas, Dimitris. "Database-friendly random projections." *In Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pp. 274-281. (2001).

[29]https://github.com/getcrest/AutoOut/tree/master/app/outlier_treatment/datasets/csv

[30] https://github.com/jbrownlee/Datasets/tree/master

[31] https://archive.ics.uci.edu/dataset/94/spambase

[32] https://github.com/GuansongPang/ADRepository-Anomaly-detection-datasets/tree/main/numerical%20data/DevNet%20datasets.

[33]http://www.svcl.ucsd.edu/projects/anomaly/dataset.htm

[34] https://archive.ics.uci.edu/dataset/171/madelon

[35] P. Rambaud and et.al. "Binary classification vs. anomaly detection on imbalanced tabular medical datasets." *In 2023 Congress in Computer Science, Computer Engineering, & Applied Computing (CSCE)*, pp. 01-05. IEEE, (2023).

[36] F. Melo. “Area under the ROC Curve.” *Encyclopedia of systems biology*, (2013).

[37] Brzezinski, Dariusz. "Random Similarity Isolation Forests." *arXiv preprint arXiv*:2502.19122 (2025).

جنگل ایزوله چرخشی گسسته در ابعاد بالا

وحیده منعمی زاده و کوروش کیانی*

دانشکده برق و کامپیوتر، دانشگاه سمنان، سمنان، ایران.

ارسال ۲۰۲۵/۰۳/۰۶؛ بازنگری ۲۰۲۵/۰۵/۲۶؛ پذیرش ۲۰۲۵/۰۶/۱۹

چکیده:

تشخیص ناهنجاری در حوزه‌هایی مانند امنیت سایبری، مدیریت ریسک مالی و پایش سلامت حیاتی و کاربردی می‌باشد، اما به دلیل پیچیدگی مجموعه داده‌های با اندازه بزرگ، داده‌های با ابعاد بالا و داده‌های بدون برچسب، همچنان چالش‌برانگیز باقی مانده است. این مقاله به بررسی رویکردهای مبتنی بر درخت تصمیم برای مقیاس‌پذیری، تفسیرپذیری و استحکام آنها با این شرایط می‌پردازد. اگرچه روش‌هایی مانند جنگل جداسازی (iForest) و جنگل جداسازی توسعه‌یافته (EIF) به‌طور گسترده مورد استفاده قرار می‌گیرند، اغلب در تشخیص ناهنجاری‌ها و جداسازی ناهنجاری‌ها از داده‌های عادی شکست می‌خورند و گاهی اوقات منجر به ایجاد «خوشه‌های شبح» می‌شوند که این امر نامطلوب می‌باشد. برای رفع این کاستی‌ها، در مطالعه پیشین [۱]، جنگل جداسازی چرخشی (RIF) را معرفی کردیم که با اعمال چرخش‌های تصادفی در فضای ویژگی‌ها، دقت تشخیص را بهبود بخشید. در ادامه این کار، «جنگل جداسازی چرخشی گسسته» (DRIF) را پیشنهاد می‌دهیم که یک رمزگذار خودکار برای کاهش ابعاد و همچنین از توزیع احتمال گسسته برای مدل‌سازی کارآمدتر ماتریس چرخش تصادفی استفاده می‌نماید. استفاده از رمزگذار خودکار، یادگیری بازنمایی را با استخراج ساختارهای پیچیده در داده‌ها بهبود می‌بخشد، در حالی‌که توزیع گسسته با حفظ معیار ROC-AUC، زمان اجرا را کاهش می‌دهد. نتایج آزمایش‌ها روی مجموعه‌داده‌های مصنوعی و واقعی نشان می‌دهد که DRIF به‌طور مداوم از نظر هر دو معیار ROC-AUC و سرعت اجرا، عملکرد بهتری نسبت به iForest، EIF و RIF دارد. این یافته‌ها، DRIF را به‌عنوان چارچوبی مقیاس‌پذیر، کارآمد و دقیق برای تشخیص ناهنجاری بدون ناظر در محیط‌های چندبُعدی معرفی می‌کنند.

کلمات کلیدی: تشخیص ناهنجاری، درخت‌های تصمیم‌گیری، جنگل‌های ایزوله چرخشی، داده‌های با ابعاد بالا در مقیاس بزرگ، یادگیری بدون نظارت، رمزگذار خودکار.