Advances in Science and Technology Research Journal

Advances in Science and Technology Research Journal, 2027, 21(1), 545–556
https://doi.org/10.12913/22998624/236910
ISSN 2299-8624, License CC-BY 4.0

Received: 2026.08.24
Accepted: 2026.11.15
Published: 2026.12.01

Dissimilarity-generalized minimum spanning tree-based isolation forest

Łukasz Gałka[1*]

[1] Department of Computer Science, Lublin University of Technology, ul. Nadbystrzycka 36B, 20-618 Lublin, Poland
E-mail: l.galka@pollub.pl

ABSTRACT

Anomaly detection is one of the major challenges in modern computer systems. The collection of increasingly large amounts of data creates a growing need for unsupervised techniques. In this article, the dissimilarity-generalized minimum spanning tree-based isolation forest (DG-MSTIF) was proposed. The method is based on constructing forests of isolation trees formed using a minimum spanning tree algorithm. One potential limitation of the baseline method is its exclusive use of the Euclidean distance, which may not be optimal for all datasets, particularly in higher-dimensional data spaces. Therefore, a generalization of the method was introduced in which different dissimilarity measures were applied, including Euclidean, Manhattan, Chebyshev, Minkowski, cosine, correlation, and Canberra. The measured area under the precision–recall curve (PR AUC) values showed that the use of appropriate dissimilarity measures can improve detection effectiveness. In addition, execution time measurements made it possible to compare the computational performance depending on the selected measure. The study indicated that an appropriate choice of dissimilarity measure can lead to improved detection performance.

Keywords: anomaly detection, isolation forest, dissimilarity measures, outlier detection, minimum spanning tree.

INTRODUCTION

Anomaly detection is one of the key elements of modern computer systems. These techniques are used to detect values that differ from other elements. For example, they are applied in cybersecurity, image analysis, network traffic analysis, and other areas. In particular, the lack of labels in the data, difficult access to obtaining them, or the high cost of acquiring them forces developers to use unsupervised anomaly detection algorithms.

Various types of algorithms are used for anomaly detection. These range from simple solutions based on *k*-nearest neighbor or local outlier factor to more complex approaches [1]. Techniques are often combined to achieve a convergence effect. Based on data distributions or the characteristics of the analyzed datasets, solutions specialized for a specific problem are developed [2]. In particular, ensembles composed of weak detectors or ensembles composed of diverse operating paradigms are constructed. The main challenges faced by these approaches include, above all, data dimensionality, irregular groups, specific cases, and changing time-series data [3]. Often, the appropriate encoding of samples itself increases detection accuracy. To extract more complex anomalies, transformations of the sample space are applied, including random transformations, deep learning-based transformations, normalization, standardization, and even attribute reduction. In the case of time series, the problem is slightly different, i.e., the data must be analyzed in a specific order [4]. Such analyses are particularly important in research fields such as financial markets, stock market data, economics, manufacturing, healthcare, and video recording analysis [5]. Any detected deviation from the data pattern must be properly identified. Moreover, it must be handled correctly, i.e., not only correctness but

also the response time of the algorithm is of key importance here.

One of the most effective algorithms is the minimum spanning tree-based isolation forest (MSTIF) [6], which originates from the isolation forest (IF) technique [7]. The method operates in two phases: the training phase and the element evaluation phase. In the first phase, isolation trees are constructed based on the minimum spanning tree algorithm. In the second phase, the samples are evaluated. By default, the algorithm uses Euclidean distance, which may not be optimal. The suitability of a particular dissimilarity measure may depend on the characteristics of the analyzed data, including their dimensionality. Therefore, the exclusive use of Euclidean distance may not be optimal for all datasets, motivating the investigation of alternative dissimilarity measures. Consequently, the minimum spanning tree may not reflect the optimal distances between samples. Ultimately, the isolation tree may be less sensitive to changes in sample positions and deviations. Therefore, in this work, a generalization of the MSTIF method was developed, extending its potential for anomaly detection through the use of different dissimilarity measures. The main contributions of this work are:

- Development of a new algorithm that generalizes the operation of the MSTIF method,
- Development of mechanisms for using different dissimilarity measures to measure the dissimilarities between samples,
- Evaluation of the developed solution on real-world anomaly detection datasets.

RELATED WORK

In this section, the solutions related to the formation of forests composed of isolation trees were reviewed. First, the baseline MSTIF method [6] was considered. In this case, isolation trees are formed based on a minimum spanning tree. This technique allows the shapes and density of the training samples to be reflected more accurately. As a result, the splits account for more complex patterns. In the base IF [7], these splits are random and perpendicular to the feature axes. On the one hand, this is computationally efficient. However, it leads to the occurrence of so-called ghost clusters. This means that the anomaly score function indicates a low anomaly value in highly outlying regions. MSTIF made it possible to reduce the impact of ghost clusters and improve the sharpness of the scoring function at the boundaries between anomalous and normal regions.

Another solution proposed in [8] is a mechanism for rotating the partition space. This mechanism allows points to be rotated before the split. Consequently, the splits are no longer perpendicular to the feature axes. The characteristic bands visible in the algorithm score heatmaps are reduced. Projections of training samples were also applied in [9]. Specifically, the data are mapped into a kernel space. Then, Principal Component Analysis is performed to extract k principal components. The samples prepared in this manner are then subjected to isolation mechanisms.

To protect the algorithm against another undesirable phenomenon, i.e., ghost inter-clusters, the elements used to form the forest are first rotated using random rotation matrices and QR decomposition [10]. This makes it possible to protect the approach against the occurrence of bands in score maps. This works even before the forest formation stage is started. Other techniques take into account not only the importance of anomalies but also the structure of regular points [11]. This approach leads to better recognition and identification of both globally and locally anomalous points.

In [12], the authors introduce layers and splits based on them. The method divides the sample space into subspaces. Consequently, it leads to more precise detection of local outliers. In [13], a random projection of the data is introduced to transform them into a new space. Then an isolation forest is applied. This is intended to protect the method against the so-called ghost clusters visible in the baseline method. Similar effects can also be achieved by transforming the space using multiple random neural networks, thereby obtaining random projections of the data [14].

Methods for excluding sample attributes based on SHapley Additive exPlanations (SHAP) values should also be mentioned [15]. In this case, some attributes are excluded from the training dataset. In addition, submodels are created, and their results are finally aggregated based on weights obtained from SHAP values.

Techniques combining IF with other algorithms are attracting growing interest. A representative example is the combination of IF with Large Language Models (LLMs) [16]. Samples are transformed into natural-language explanations, which makes it possible to improve confidence in their anomalousness. Consequently, the anomaly score becomes more transparent.

It is also worth mentioning studies that introduce fuzzy logic and fuzzy sets into the isolation mechanism. In [17], fuzzy aggregation was applied to factors related to distance and to a factor associated with the weight of a minimum spanning tree edge. This algorithm allows the appropriate class to be assigned to a sample more precisely. It also introduces mechanisms for interpreting the score values, which is particularly important in an unsupervised setting. Another improvement, presented in [18], considers a fuzzy traversal of a sample through isolation trees. The paths are then appropriately weighted and aggregated so that the scoring function can determine the uncertainty levels of the splits themselves.

Another study [19] focused on optimal calibration of the decision threshold. Specifically, in an unsupervised setting, the contamination of anomalous samples is in fact unknown. The above work indicated the possibility of determining the threshold through the analysis of sample distributions performed using various techniques. As a result, the ranking itself remains unchanged, while the detection threshold can be adjusted more precisely, improving effectiveness.

IF-based methods are also successfully utilized for streaming data [20]. In such cases, it is important for the model to remain up to date with the current data. Therefore, a scheme using subtrees and tree reconstruction strategies was introduced.

Finally, it should be emphasized that both effectiveness and interpretability are major research topics. In [21], prediction explanations were applied for industrial applications. The article introduces feature importance analysis and a new measure that uses feature importance to validate interpretability.

To summarize, many concepts for improving isolation forests have been presented. An interested reader may refer to a more comprehensive survey in [22]. Although MST-based isolation approaches have already been proposed, the influence of the selected dissimilarity measure on their operation and effectiveness has not been systematically investigated. In contrast to the original MSTIF [6], which uses the Euclidean distance, the proposed DG-MSTIF generalizes the method by allowing different dissimilarity measures to be employed during both MST construction and sample evaluation.

DISSIMILARITY-GENERALIZED MINIMUM SPANNING TREE-BASED ISOLATION FOREST

In this section, a generalization of the MSTIF technique called dissimilarity-generalized minimum spanning tree-based isolation forest (DG-MSTIF) is introduced. The method operates in two phases. The first phase involves the formation of a forest of isolation trees. The second phase includes the evaluation of samples. The general workflow of the proposed DG-MSTIF method is presented in Figure 1.

Training phase

The isolation forest is formed in the following manner. Each tree in the forest is constructed according to the same procedure by randomly selecting a subset of samples. Next, the samples are standardized as follows:

$$\mu_j = \frac{1}{\psi}\sum_{i=1}^{\psi} x_{ij} \tag{1}$$

and

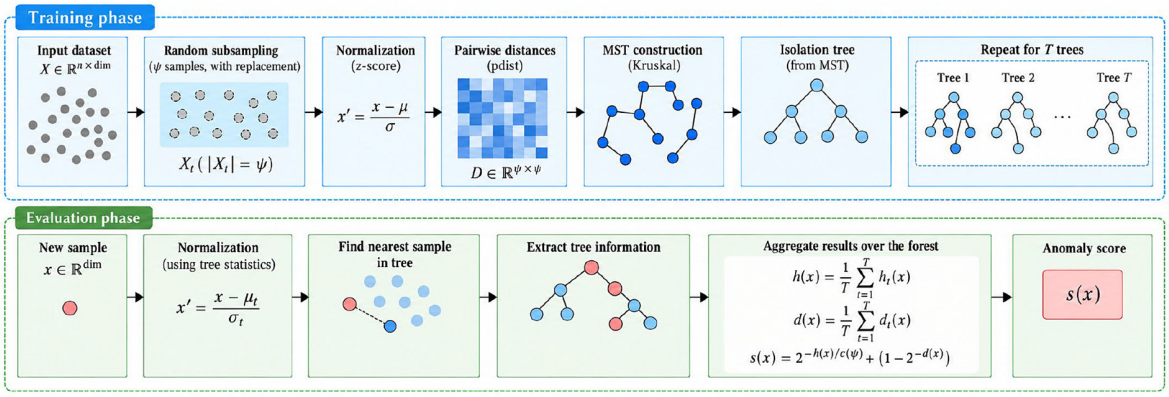


Figure 1. General workflow of the proposed DG-MSTIF method, including the training and evaluation phases

$$\sigma_j = \sqrt{\frac{1}{\psi}\sum_{i=1}^{\psi}\left(x_{ij} - \mu_j\right)^2} \tag{2}$$

Consequently,

$$\hat{x}_{ij} = \frac{x_{ij} - \mu_j}{\sigma_j + \epsilon}, \epsilon = 10^{-8} \tag{3}$$

where: μ_j and σ_j denote the mean and standard deviation of the j-th attribute, respectively. The $\in$ preserves the algorithm against division by 0. This transformation ensures comparable scales between the attributes.

In the next step, a list of edges used to construct the minimum spanning tree (MST) is prepared. At this point, the algorithm is extended with an additional parameter specifying the dissimilarity measure used for this purpose. For each pair of standardized samples $\hat{x}_i$ and $\hat{x}_j$, where $i < j$, the dissimilarity is calculated as

$$d_{ij} = d_M\left(\hat{x}_i, \hat{x}_j\right), \tag{4}$$

where: M denotes the selected dissimilarity measure.

The set of all candidate edges is defined as

$$E = \left\{\left(i, j, d_{ij}\right): 1 \le i < j \le \psi\right) \tag{5}$$

The calculated dissimilarity values constitute the weights of the candidate MST edges. The edges are then sorted in ascending order according to their weights:

$$d_{e_1} \le d_{e_2} \le \cdots \le d_{e_{|E)}} \tag{6}$$

Based on the ordered set of edges, the MST is constructed using Kruskal's algorithm:

$$MST_{Tree} = Kruskal(V, E) \tag{7}$$

where: V denotes the set of training samples treated as graph vertices, and MST_{Tree} is the resulting minimum spanning tree.

It should be noted that the considered dissimilarity measures are not required to satisfy all metric properties. In particular, cosine and correlation dissimilarities may not satisfy the triangle inequality. However, this does not preculde their application in the proposed approach, since Kruskal's algorithm operates on weighted graph edges and requires only their ordering according to the assigned weights. Therefore, the MST and the subsequent isolation-tree construction remain well defined also for non-metric dissimilarity measures.

Based on the MST constructed in this manner, an isolation tree is built. Specifically, at the beginning, a node constituting an individual tree is created for each sample. Next, the trees prepared in this manner are merged according to the order in which the nodes are connected during MST construction using Kruskal's algorithm. More precisely, merging two subtrees during MST formation creates a root node that connects these two subtrees in the constructed isolation tree. According to this procedure, these steps terminate when the MST is completed. Consequently, the isolation tree contains all subtrees, where the leaf nodes correspond to the considered samples and the internal nodes represent MST merges. The isolation tree constructed in this manner is a binary tree. Leaf nodes located at greater depths represent more regular samples, whereas leaf nodes located at lower depths represent more anomalous samples.

Next, in order to speed up the evaluation phase, the depth of each node in the isolation tree is calculated. Let $p(v)$ denote the parent of node v. The depth of a node can be defined recursively as

$$h(v) = \begin{cases} 0, & p(v) = v, \\ h\left(p(v)\right) + 1, & p(v) \ne v. \end{cases} \tag{8}$$

Equivalently, $h(v)$ corresponds to the number of parent transitions required to reach the root of the tree. The trees prepared in this manner are stored in memory for use during the second evaluation phase. It is important to highlight that this phase is conducted only once. The pseudocode for the training phase is presented in Algorithm 1.

Evaluation phase

The second phase, i.e., sample evaluation, is performed independently for each sample. Therefore, the computations can be easily parallelized. Each evaluated sample is processed independently by every isolation tree constructed during the training phase. First, for each tree, the evaluated sample is standardized using the mean and standard deviation values calculated during the training phase, according to Eq. (3).

Algorithm 1. Training phase of DG-MSTIF

Input: dataset X, number of trees T, subsample size ψ, dissimilarity measure δ, and its parameters
Output: forest F composed of T MST-based isolation trees
1. Initialize an empty forest F.
2. **For** t = 1, 2, ..., T:
Randomly select ψ samples from X with replacement to form X_t.
Compute the per-feature mean μ_t and standard deviation σ_t of X_t.
Standardize the subsample: X'_t ← (X_t − μ_t)/(σ_t + ε).
Compute all pairwise dissimilarities δ(x'_i, x'_j) between samples in X'_t.
Create the complete weighted graph using the calculated dissimilarities as edge weights.
Sort the candidate edges in ascending order of their weights.
Construct the minimum spanning tree using Kruskal's algorithm.
Whenever two components are merged, create a new internal isolation-tree node and store the corresponding merge dissimilarity.
Compute the depth of each leaf node corresponding to a training sample.
Store X'_t, μ_t, σ_t, the parent structure, merge dissimilarities, and leaf depths in tree t.
Append tree t to F.
3. Return F.

Let $\hat{x}$ denote the standardized evaluated sample and let $\hat{x}_i^{(t)}$ denote the *i*-th standardized training sample forming the *t*-th isolation tree. The dissimilarities between the evaluated sample and all leaf nodes of the tree are calculated using the dissimilarity measure *M*:

$$d_i^{(t)}(x) = d_M\left(\hat{x}, \hat{x}_i^{(t)}\right), i = 1, \dots, \psi \tag{9}$$

Next, the leaf node corresponding to the training sample closest to the evaluated sample is determined as

$$i_t^\star(x) = arg \min_{i \in \{1,\dots,\psi\}} d_i^{(t)}(x) \tag{10}$$

For the selected leaf node, two values are obtained. The first one is the dissimilarity between the evaluated sample and its nearest training sample:

$$d_t(x) = d_{i_t^\star(x)}^{(t)}(x) \tag{11}$$

whereas the second one is the previously calculated depth of this leaf node in the isolation tree:

$$h_t(x) = h_t\left(i_t^\star(x)\right) \tag{12}$$

After processing the sample using all *T* isolation trees, the average depth and average dissimilarity are calculated as

$$\bar{h}(x) = \frac{1}{T}\sum_{t=1}^{T} h_t\,(x) \tag{13}$$

and

$$\bar{d}(x) = \frac{1}{T}\sum_{t=1}^{T} d_t\,(x) \tag{14}$$

To normalize the average depth, the correction factor $c(\psi)$ is calculated as

$$c(\psi) = \begin{cases} 0, & \psi \le 1, \\ 2(ln(\psi - 1) + \gamma) - \dfrac{2(\psi - 1)}{\psi}, & \psi > 1, \end{cases} \tag{15}$$

where: $\gamma \approx 0.5772156649$ is the Euler–Mascheroni constant.

Finally, the anomaly score of sample *x* is calculated as

$$s(x) = 2^{-\frac{\bar{h}(x)}{c(\psi)}} + \left(1 - 2^{-\bar{d}(x)}\right) \tag{16}$$

The anomaly score is bounded by 0 and 2, which is an intended extension of the original IF score range. The depth component increases when a sample is isolated at a smaller depth, whereas the dissimilarity component increases monotonically with the average dissimilarity. Importantly, the dissimilarity component is calculated in the standardized feature space according to Eq. (3). The exponential transformation maps the resulting non-negative dissimilarity values to the interval [0,1). Consequently, a higher anomaly score indicates a greater likelihood that the evaluated sample is anomalous. One can note that the selected dissimilarity measure affects both components of the final anomaly score. In addition to

determining the dissimilarity to the nearest training sample, it determines the MST edge weights and may consequently change the structure of the isolation tree and the corresponding leaf depths. The pseudocode for the evaluation phase is presented in Algorithm 2.

Although the input attributes are standardized, different dissimilarity measures may still produce values on different numerical scales. Consequently, the absolute anomaly-score values obtained using different dissimilarity measures should not be interpreted as directly comparable. No additional measure-specific normalization is applied in this study, since each DG-MSTIF configuration is evaluated independently using PR AUC.

Computational complexity

For a single isolation tree constructed from ψ samples, the calculation of all pairwise dissimilarities requires $O(\psi^2)$ dissimilarity evaluations. The complete graph contains $E = \frac{\psi(\psi-1)}{2}$ candidate edges, and the MST construction using Kruskal's algorithm requires $O(ElogE)$ operations, which corresponds to $O(\psi^2 log\psi)$. For the evaluation phase, each evaluated sample is compared with ψ training samples in each tree, resulting in $O(\psi)$ dissimilarity evaluations per tree. Therefore, for a forest composed of T trees, the corresponding complexities are $O(T\psi^2 log\psi)$ for training and $O(T\psi)$ per evaluated sample for evaluation. The computational cost of each dissimilarity evaluation additionally depends on the number of attributes and the selected dissimilarity measure, which explains the observed influence of both data dimensionality and measure choice on the execution time.

NUMERICAL EXPERIMENTS

In this section, the results of numerical experiments confirming the effectiveness of DG-MSTIF are presented. Since the Euclidean configuration corresponds to the baseline MSTIF, it is treated as the reference configuration in order to isolate the influence of the dissimilarity measure. The algorithm can be configured with different non-negative dissimilarity measures. In the experiments, the following dissimilarity measures are considered: Euclidean (EUC), Manhattan (MAN), Chebyshev (CHEB), Minkowski with $p = 1.5$ (M1.5), $p = 3$ (M3), and $p = 4$ (M4), cosine (COS), correlation (COR), and Canberra (CAN). Moreover, the experiments were conducted on a laptop equipped with an Intel Core i7-10750H processor and 32 GB of RAM. Finally, DG-MSTIF was configured with $T = 100$ isolation trees and $\psi = 256$ samples used to construct each tree.

For each isolation tree, samples are selected independently using uniform random sampling with replacement. Pairwise dissimilarities during the training phase are calculated using the `pdist` function, whereas dissimilarities during sample evaluation are calculated using `cdist` from the SciPy library. Candidate MST edges are sorted in ascending order by weight. No additional tie-breaking criterion is applied. Therefore, edges with equal weights are processed in the order returned by the sorting procedure. Zero-weight

Algorithm 2. Evaluation phase of DG-MSTIF

Input: evaluated samples X_eval, trained forest F, subsample size ψ **Output:** anomaly score s(x) for each evaluated sample x
1. Compute the correction factor c(ψ). 2. **For** each evaluated sample x: Initialize H ← 0 and D ← 0. **For** each tree t ∈ F: Standardize x using the statistics of tree t: x'_t ← (x − μ_t)/(σ_t + ε). Compute the dissimilarity between x'_t and every standardized training sample stored in tree t. Find the nearest training sample i* corresponding to the minimum dissimilarity. Set h_t(x) to the stored depth of leaf i* and d_t(x) to the corresponding minimum dissimilarity. **If** d_t(x) = 0, traverse the ancestors of leaf i* until the first non-zero merge dissimilarity is found and use it as d_t(x). Accumulate H ← H + h_t(x) and D ← D + d_t(x). Compute the average depth $\bar{h}(x)$ ← H/T and average dissimilarity $\bar{d}(x)$ ← D/T. Compute s(x) ← 2^(−$\bar{h}$(x)/c(ψ)) + [1 − 2^(−$\bar{d}$(x))]. 3. Return the anomaly scores for all evaluated samples.

edges, which may occur when the same sample is selected more than once, are retained and processed by Kruskal's algorithm. The cosine, correlation, and Canberra dissimilarities are calculated directly using their SciPy implementations after z-score standardization, without additional measure-specific preprocessing.

Dataset statistics

The comparisons include results obtained by DG-MSTIF on a diverse set of real-world anomaly detection datasets from [23]. The basic statistics of the datasets are presented in Table 1. The datasets were selected to ensure diversity in terms of the number of samples, data dimensionality, and the percentage of anomalies.

Performance results

In the experiments, the entire dataset is used for both model training and sample scoring. The models are trained in a fully unsupervised manner. For each tree, samples are selected independently with replacement. The method is non-deterministic, and each of the 50 repetitions uses a different random seed. This results in different sample subsets and consequently different forests.

First, the effectiveness of different dissimilarity measures was analyzed. To obtain robust results, the experiments were repeated 50 times, and the reported values represent the mean and standard deviation. The PR AUC results are presented in Table 2. As can be readily observed, the Euclidean measure may not be the optimal choice for the analyzed datasets. Interestingly, the mean and median values obtained for the Manhattan measure achieved the best results. This shows that generalizing the dissimilarity measure can lead to improved effectiveness. Additionally, although the Euclidean measure is widely used, it does not demonstrate equally strong effectiveness in the problem of unsupervised anomaly detection. The similarity between the COS and COR results may be partly attributed to z-score standardization, although the two measures are not equivalent, as correlation additionally centers the compared vectors with respect to their individual means. It should be emphasized that,in a fully unsupervised setting, the best-performing dissimilarity measure for a particular dataset cannot be selected directly based on the target labels. Therefore, the per-dataset best results should be interpreted as an estimate of the upper-bound performance achievable. The main purpose of this comparison is to demonstrate the potential influence of the dissimilarity measure on the MSTIF effectiveness. Moreover, it shows that the default Euclidean measure is

Table 1. Dataset statistics

Dataset	Observations	Attributes	Anomalies
Annthyroid	7200	6	534 (7.42 %)
Cardio	1831	21	176 (9.61 %)
Cardiotocography	2114	21	466 (22.04 %)
Fault	1941	27	673 (34.67 %)
Glass	214	9	9 (4.21 %)
Hepatitis	80	19	13 (16.25 %)
InternetAds	1966	1555	368 (18.72 %)
Ionosphere	351	33	126 (35.90 %)
Landsat	6435	36	1333 (20.71 %)
Letter	1600	32	100 (6.25 %)
Optdigits	5216	64	150 (2.88 %)
SpamBase	4207	57	1679 (39.91 %)
Stamps	340	9	31 (9.12 %)
Vertebral	240	6	30 (12.50 %)
Wilt	4819	5	257 (5.33 %)
Wine	129	13	10 (7.75 %)
WPBC	198	33	47 (23.74 %)
Yeast	1484	8	507 (34.16 %)

Table 2. PR AUC results [%] for different dissimilarity measures, reported as mean values with standard deviations

Dataset	Dissimilarity measure								
	EUC	MAN	CHEB	M1.5	M3	M4	COS	COR	CAN
Annthyroid	29.37 (0.88)	32.21 (0.88)	27.56 (0.85)	30.23 (0.89)	28.70 (0.98)	28.55 (0.99)	35.32 (1.79)	18.52 (1.27)	36.03 (2.18)
Cardio	45.12 (0.59)	50.62 (0.72)	49.06 (0.76)	46.04 (1.12)	46.61 (0.60)	47.64 (0.67)	11.27 (0.48)	9.05 (0.62)	7.56 (0.10)
Cardiotocography	34.30 (0.28)	35.95 (0.41)	34.75 (0.24)	34.42 (0.28)	34.18 (0.29)	34.44 (0.25)	16.75 (0.19)	15.22 (0.15)	15.28 (0.12)
Fault	50.47 (0.44)	40.30 (0.45)	50.64 (0.33)	48.84 (0.50)	51.01 (0.38)	50.95 (0.29)	51.72 (0.67)	50.87 (0.91)	43.05 (0.61)
Glass	9.72 (0.23)	9.12 (0.26)	9.85 (0.14)	9.57 (0.22)	9.82 (0.19)	9.84 (0.17)	22.37 (1.17)	21.45 (1.01)	9.76 (0.57)
Hepatitis	24.97 (0.36)	21.12 (0.16)	21.18 (0.44)	23.21 (0.33)	25.09 (0.42)	24.03 (0.56)	25.47 (0.33)	26.44 (0.25)	17.72 (0.19)
InternetAds	39.16 (1.50)	41.78 (1.37)	20.49 (1.59)	40.52 (1.31)	27.54 (1.48)	26.39 (2.01)	34.01 (1.36)	28.33 (0.78)	47.02 (0.84)
Ionosphere	93.77 (0.05)	87.56 (0.23)	93.99 (0.04)	92.75 (0.06)	94.19 (0.02)	94.27 (0.02)	88.76 (0.27)	91.35 (0.22)	58.30 (0.54)
Landsat	22.57 (0.13)	23.34 (0.25)	21.42 (0.11)	22.77 (0.16)	22.27 (0.14)	22.07 (0.13)	24.83 (0.25)	41.54 (0.49)	23.49 (0.19)
Letter	16.93 (0.37)	10.22 (0.18)	23.39 (0.82)	13.33 (0.27)	20.97 (0.38)	22.84 (0.46)	29.54 (0.93)	33.86 (1.06)	11.31 (0.27)
Optdigits	5.70 (0.69)	29.46 (2.05)	1.97 (0.04)	13.84 (1.09)	2.74 (0.18)	2.33 (0.08)	10.65 (1.82)	11.31 (1.68)	92.87 (2.69)
SpamBase	42.36 (0.41)	52.41 (0.38)	38.70 (0.57)	45.20 (0.56)	40.23 (0.36)	39.64 (0.60)	47.57 (0.55)	44.15 (0.71)	63.07 (0.27)
Stamps	28.39 (0.32)	28.61 (0.46)	28.98 (0.70)	27.97 (0.48)	28.73 (0.45)	28.82 (0.36)	10.25 (0.23)	9.23 (0.16)	9.41 (0.18)
Vertebral	8.58 (0.04)	8.56 (0.05)	8.92 (0.03)	8.56 (0.03)	8.62 (0.04)	8.67 (0.04)	17.43 (0.49)	10.62 (0.16)	11.51 (0.18)
Wilt	5.27 (0.07)	5.39 (0.04)	5.07 (0.05)	5.31 (0.05)	5.20 (0.06)	5.17 (0.05)	8.88 (0.89)	30.00 (1.47)	7.84 (0.18)
Wine	11.23 (0.28)	11.86 (0.27)	12.81 (0.29)	9.88 (0.22)	13.54 (0.25)	14.00 (0.43)	11.00 (0.44)	8.74 (0.33)	4.85 (0.05)
WPBC	21.41 (0.08)	22.73 (0.15)	21.13 (0.11)	21.67 (0.11)	21.29 (0.10)	21.29 (0.08)	26.59 (0.31)	23.53 (0.29)	29.27 (0.31)
Yeast	29.69 (0.14)	29.54 (0.10)	29.80 (0.11)	29.56 (0.11)	29.70 (0.11)	29.72 (0.09)	35.25 (0.24)	32.93 (0.21)	30.39 (0.14)
Mean	28.83 (0.38)	30.04 (0.47)	27.76 (0.40)	29.09 (0.43)	28.36 (0.36)	28.37 (0.41)	28.20 (0.69)	28.17 (0.65)	28.82 (0.53)
Median	26.68 (0.30)	29.04 (0.26)	22.40 (0.27)	25.59 (0.28)	26.31 (0.27)	25.21 (0.27)	25.15 (0.48)	24.99 (0.56)	20.61 (0.23)

not necessarily optimal. The development of a fully unsupervised procedure for selecting an appropriate dissimilarity measure constitutes a separate research problem.

To further compare the considered dissimilarity measures across all datasets, the Friedman test was applied with a significance level of $\alpha = 0.05$. The test yielded $p = 0.413$, indicating that no statistically significant differences between the considered dissimilarity measures were identified. Nevertheless, differences in their average ranks were observed, as illustrated by the critical difference diagram in Figure 2. Although the average ranks indicate differences in the overall performance of the considered measures, these differences should be interpreted as descriptive because the Friedman test did not indicate statistical significance.

The obtained results can be directly related to the original MSTIF method [6], since the EUC configuration of DG-MSTIF corresponds to the original formulation based on Euclidean distance. Therefore, the EUC results reported in Table 2 constitute a direct reference for evaluating the influence of alternative dissimilarity measures. The mean PR AUC increased from 28.83% for EUC to 30.04% for MAN, while the median increased from 26.68% to 29.04%. This indicates that replacing the fixed Euclidean distance with another dissimilarity measure can improve the

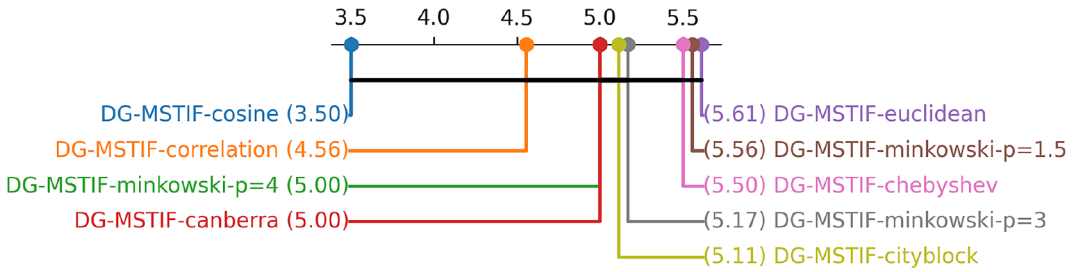


Figure 2. Critical difference diagram showing the average ranks of the considered dissimilarity measures across all datasets

effectiveness of MSTIF for certain data distributions. At the same time, no single dissimilarity measure dominated across all datasets, and the Friedman test did not indicate statistically significant global differences. Thus, the results should be interpreted as evidence that the suitability of the dissimilarity measure is dataset-dependent rather than as evidence of the universal superiority of one particular measure.

The proposed approach differs from previously introduced extensions of Isolation Forest that modify the partitioning mechanism through rotations, projections, feature selection, or alternative traversal and scoring mechanisms. In DG-MSTIF, the general structure of the MSTIF approach is preserved, whereas the dissimilarity used during MST construction and sample evaluation is generalized. In relation to the original MSTIF [6], the

Table 3. Training time [ms] for different dissimilarity measures, reported as mean values with standard deviations

Dataset	Dissimilarity measure								
	EUC	MAN	CHEB	M1.5	M3	M4	COS	COR	CAN
Annthyroid	392 (22)	409 (41)	370 (7)	982 (8)	990 (7)	994 (12)	396 (15)	405 (17)	420 (17)
Cardio	426 (18)	449 (13)	381 (30)	2076 (65)	2103 (86)	2134 (10)	428 (12)	424 (6)	487 (8)
Cardiotocography	422 (19)	419 (11)	384 (11)	2157 (22)	2101 (96)	2125 (18)	427 (9)	429 (8)	481 (9)
Fault	453 (14)	436 (20)	435 (7)	2897 (234)	2757 (25)	2717 (55)	422 (12)	428 (13)	517 (5)
Glass	408 (11)	397 (15)	351 (3)	1165 (33)	1238 (15)	1130 (26)	391 (4)	414 (9)	435 (13)
Hepatitis	399 (13)	391 (11)	301 (5)	1808 (16)	1814 (11)	1815 (9)	393 (11)	404 (10)	419 (20)
InternetAds	4082 (317)	5423 (326)	4616 (73)	51018 (2464)	48552 (1713)	50368 (374)	4670 (203)	5307 (47)	9080 (72)
Ionosphere	549 (29)	489 (19)	470 (32)	3374 (194)	3275 (146)	3435 (93)	472 (22)	442 (24)	560 (25)
Landsat	508 (22)	510 (66)	485 (34)	3628 (145)	3581 (28)	3503 (84)	493 (27)	494 (46)	645 (135)
Letter	462 (18)	475 (16)	392 (14)	3148 (34)	3048 (133)	3069 (134)	431 (10)	471 (15)	546 (19)
Optdigits	559 (50)	537 (47)	469 (13)	5008 (40)	5027 (36)	4919 (102)	506 (21)	531 (19)	788 (82)
SpamBase	500 (19)	530 (32)	461 (26)	3769 (79)	3651 (243)	3839 (189)	484 (22)	501 (28)	672 (28)
Stamps	431 (15)	407 (17)	409 (17)	1197 (39)	1252 (57)	1265 (39)	408 (15)	421 (12)	480 (42)
Vertebral	421 (13)	415 (7)	420 (9)	969 (19)	1036 (10)	1033 (10)	411 (7)	409 (8)	412 (14)
Wilt	460 (22)	440 (11)	460 (37)	948 (20)	933 (13)	922 (57)	424 (20)	438 (34)	434 (17)
Wine	416 (17)	397 (18)	391 (16)	1623 (10)	1635 (12)	1547 (84)	406 (25)	418 (11)	451 (15)
WPBC	478 (13)	492 (8)	528 (81)	3343 (80)	3306 (139)	3333 (104)	452 (7)	473 (53)	531 (33)
Yeast	433 (19)	426 (18)	390 (16)	1070 (32)	1048 (40)	1112 (124)	421 (15)	440 (19)	451 (44)
Mean	655 (36)	724 (39)	651 (24)	5010 (196)	4853 (156)	4959 (85)	669 (25)	714 (21)	989 (33)
Median	443 (18)	438 (17)	414 (16)	2117 (36)	2102 (38)	2130 (70)	426 (15)	434 (16)	484 (20)

Table 4. Evaluation time [ms] for different dissimilarity measures, reported as mean values with standard deviations

Dataset	Dissimilarity measure								
	EUC	MAN	CHEB	M1.5	M3	M4	COS	COR	CAN
Annthyroid	1172 (59)	1073 (57)	1039 (16)	32654 (645)	33600 (61)	33555 (62)	1309 (19)	1316 (81)	1940 (26)
Cardio	696 (29)	755 (11)	688 (37)	24085 (644)	25299 (598)	24556 (71)	628 (7)	618 (57)	1530 (10)
Cardiotocography	751 (13)	824 (16)	864 (13)	28654 (70)	28008 (645)	28517 (79)	726 (19)	767 (23)	1776 (37)
Fault	870 (56)	1004 (48)	964 (14)	35255 (869)	35205 (671)	35105 (552)	830 (27)	844 (31)	2518 (58)
Glass	46 (3)	42 (7)	41 (5)	1336 (17)	1418 (14)	1312 (31)	47 (7)	58 (9)	84 (4)
Hepatitis	32 (3)	32 (3)	32 (3)	913 (15)	922 (11)	925 (9)	29 (2)	36 (1)	59 (4)
InternetAds	57005 (2789)	78670 (660)	65693 (1511)	760829 (1044)	761764 (1414)	766153 (7506)	72628 (315)	77438 (284)	124628 (681)
Ionosphere	228 (44)	226 (11)	200 (14)	7984 (206)	7935 (288)	7905 (275)	177 (11)	185 (12)	444 (29)
Landsat	3898 (407)	4514 (188)	4489 (491)	159862 (572)	156832 (1345)	158522 (3153)	3633 (145)	3563 (228)	8743 (598)
Letter	808 (35)	960 (31)	908 (64)	34175 (49)	33243 (1038)	33753 (532)	675 (19)	733 (37)	1953 (20)
Optdigits	5061 (158)	6128 (343)	6130 (249)	182390 (797)	180799 (3258)	180288 (300)	4726 (166)	5027 (396)	12861 (195)
SpamBase	3333 (103)	4516 (339)	4030 (158)	107778 (354)	107307 (898)	105316 (2822)	3573 (223)	3689 (150)	8744 (309)
Stamps	72 (7)	66 (8)	65 (6)	2254 (25)	2297 (69)	2299 (60)	73 (5)	85 (10)	138 (14)
Vertebral	42 (4)	40 (4)	37 (2)	1164 (50)	1194 (13)	1204 (11)	43 (1)	54 (7)	64 (5)
Wilt	797 (26)	653 (25)	643 (46)	19487 (633)	19972 (69)	19920 (356)	837 (35)	892 (64)	1284 (158)
Wine	41 (6)	34 (1)	36 (2)	1265 (35)	1285 (17)	1225 (59)	38 (2)	48 (5)	76 (7)
WPBC	106 (3)	129 (6)	124 (23)	4627 (94)	4687 (79)	4576 (136)	101 (4)	124 (35)	243 (26)
Yeast	287 (10)	248 (19)	253 (13)	7858 (22)	7669 (131)	7490 (370)	296 (15)	318 (21)	518 (112)
Mean	4180 (209)	5551 (99)	4791 (148)	78476 (341)	78302 (590)	78479 (910)	5020 (57)	5322 (81)	9311 (127)
Median	724 (28)	704 (17)	665 (15)	21786 (150)	22636 (210)	22238 (206)	652 (17)	675 (33)	1407 (28)

results demonstrate that the Euclidean distance does not necessarily provide the most suitable representation of sample relationships for every dataset. Therefore, the results complement previous research on modifications of isolation-based methods by identifying the dissimilarity measure itself as an additional factor that can influence anomaly detection effectiveness.

Runtime measurements

The second series of experiments presents the execution times of the individual phases of the algorithm. This is particularly important for applications of unsupervised anomaly detection. For this reason, the measurements were repeated 10 times, with 5 instances of the algorithm executed in each repetition, to assess execution speed. The runtime results are presented in Tables 3 and 4 for the training phase and the sample evaluation phase, respectively.

The results show a strong influence of data dimensionality on the measured execution times. This is related to the operating characteristics of DG-MSTIF. The use of the Chebyshev measure turns out to be the fastest during training, whereas Minkowski is the slowest. The substantially longer execution times observed for the Minkowski measures result from the higher computational cost of the general power operations required for their calculation. Therefore, they should be interpreted as an implementation-dependent effect rather than as an inherent increase in asymptotic complexity. It is worth

noting that these execution times are still relatively short and, in most cases, do not exceed one second per instance. The results for the sample evaluation phase show, as expected, that the execution time increases with the number of samples. However, measures such as cosine, Chebyshev, correlation, and Manhattan have the fastest execution in terms of the median value. Therefore, it seems reasonable to select dissimilarity measures that, on the one hand, improve detection performance, as shown in Table 2, and, on the other hand, provide fast execution, as demonstrated in Tables 3 and 4.

Limitations

It is also worth mentioning the identified limitations. On the one hand, for high-dimensional data, the method becomes slower. Therefore, the development of techniques for accelerating the approach even in high-dimensional spaces is identified as one of the main directions for future research. This can be achieved, for example, by applying dimensionality reduction techniques such as Principal Component Analysis. Dimensionality reduction can also be performed in a faster manner, for example, by randomly selecting attributes. This may additionally provide greater robustness of DG-MSTIF to overfitting.

Another important direction for future research is the development of an unsupervised technique for selecting an appropriate dissimilarity measure for a particular dataset. Such a selection could be performed by developing a dataset similarity measure. This would make it possible to determine which previously analyzed dataset is most similar to the evaluated one and then apply the corresponding dissimilarity measure.

CONCLUSIONS

In this study, a generalization of the MSTIF method called Dissimilarity-Generalized Minimum Spanning Tree-Based Isolation Forest was introduced. The algorithm generalizes MSTIF by using different dissimilarity measures. Extensive numerical experiments showed that changing the dissimilarity measure can improve the effectiveness of the algorithm. This is particularly important for unsupervised anomaly detection, where the algorithm learns the structure of the data without labels. Additionally, to assess the applicability of the proposed solution, the execution times of the individual phases of the algorithm were measured. The results showed that the proposed solution can be successfully applied and, in most cases, achieves execution times comparable to the default behavior of MSTIF using the Euclidean measure.

Acknowledgements

During the preparation of this work, ChatGPT was used for proofreading and language correction. The generated suggestions were reviewed and edited as needed, and full responsibility for the content of the published article is retained by the author. This work was supported by the internal grant FD-20/IT-3/047.

REFERENCES

1. Klarak J, Chen L, Cikalova U, Malık P, Andok R, Bendjus B. Defect detection in battery electrode production using supervised and unsupervised learning with laser speckle photometry data. *Adv. Sci. Technol. Res. J.* 2025; 19: 258–269. https://doi.org/10.12913/22998624/200823
2. Dymora P, Mazurek M, Lukiewicz K, Bokota K. Comparative analysis of machine learning models for multiclass anomaly detection in iot network traffic using the RT-IoT2022 Dataset. *Adv. Sci. Technol. Res. J.* 2026; 20: 45–59. https://doi.org/10.12913/22998624/213807
3. Carletti V, Foggia P, Rosa F, Vento M. Detecting malicious IoT network communication through graph neural networks in real-world conditions. *Pattern Recognit. Lett.* 2025; 189: 92–8. https://doi.org/10.1016/j.patrec.2025.01.010
4. Zamanzadeh Darban Z, Webb GI, Pan S, Aggarwal C, Salehi M. Deep learning for time series anomaly detection: a survey. *Acm Comput. Surv.* 2024; 57: 1-42. https://doi.org/10.1145/3691338
5. Kuras P, Bolanowski M, Loza M. Optimization of machine learning models for effective anomaly detection in industrial IoT systems. *Adv. Sci. Technol. Res. J.* 2026; 20:203–21. https://doi.org/10.12913/22998624/210686
6. Gałka L, Karczmarek P, Tokovarov M. Isolation forest based on minimal spanning tree. *IEEE Access* 2022; 10: 74175–86. https://doi.org/10.1109/access.2022.3190505
7. Liu FT, Ting KM, and Zhou ZH. Isolation Forest. 2008 Eighth Ieee Int. Conf. Data Min. *IEEE*, 2008: 413–22. https://doi.org/10.1109/ICDM.2008.17

8. Hariri S, Kind MC, Brunner RJ. Extended Isolation Forest. IEEE Trans. Knowl. *Data Eng*. 2021; 33: 1479–89. https://doi.org/10.1109/TKDE.2019.2947676
9. Li S, Zhang K, Duan P, Kang X. Hyperspectral Anomaly Detection with Kernel Isolation Forest. *IEEE Trans. Geosci. Remote Sens*. 2020; 58: 319–29. https://doi.org/10.1109/TGRS.2019.2936308
10. Monemizadeh V and Kiani K. Detecting Anomalies Using Rotated Isolation Forest. Data Mining and Knowledge Discovery 2025 Mar; 39: 24. https://doi.org/10.1007/s10618-025-01096-5
11. Yepmo V, Smits G, Lesot MJ, and Pivert O. Leveraging an Isolation Forest to Anomaly Detection and Data Clustering. *Data Knowl. Eng.* 2024; 151: 102302. doi: https://doi.org/10.1016/j.datak.2024.102302
12. Liu T, Zhou Z, and Yang L. Layered Isolation Forest: A Multi-Level Subspace Algorithm for Improving Isolation Forest. Neurocomputing 2024; 581: 127525. doi: https://doi.org/10.1016/j.neucom.2024.127525
13. Tan X, Yang J, and Rahardja S. Sparse Random Projection Isolation Forest for Outlier Detection. *Pattern Recognit. Lett.* 2022; 163: 65–73. https://doi.org/10.1016/j.patrec.2022.09.015
14. Gałka L. Optimized Deep Isolation Forest. *Pattern Recognit. Lett.* 2025; 197: 88–94. https://doi.org/10.1016/j.patrec.2025.07.014
15. Rachwał A, Karczmarek P, Rachwał A, Stegierski R. Isolation Forest with Exclusion of Attributes Based on Shapley Index. IEEE Access 2024; 12: 101797–813. https://doi.org/10.1109/access.2024.3432174
16. Seo J, Bang G, Kim D, and Min M. ReForest: Interpretability Meets Reasoning in Isolation Forest. Adv. Knowl. Discov. Data Min. Ed. by Wong RCW, Tong H, Lu H, Kwok J, Salim F, Song Y, Yiu ML. Singapore: Springer Nature Singapore, 2026: 628–39. https://doi.org/10.1007/978-981-92-1468-6_45
17. Gałka L. MSTIF-IFS: Isolation Forest Based on Minimal Spanning Tree and Intuitionistic Fuzzy Sets. *Pattern Recognit.* 2026; 178: 113507. https://doi.org/10.1016/j.patcog.2026.113507
18. Kim K. Fuzzy Anomaly Scores for Isolation Forest. *Appl. Soft Comput.* 2024; 166: 112193. https://doi.org/10.1016/j.asoc.2024.112193
19. Mamun AA and Herndon N. UACE-if-AD: Unsupervised Tail-Mass Calibration for Isolation Forest under Unknown Contamination. *IEEE Access* 2026; 14: 38908–26. https://doi.org/10.1109/ACCESS.2026.3671614
20. Liu JJ, Cassales GW, Liu FT, Pfahringer B,Bifet A. Streaming Isolation Forest. Adv. Knowl. Discov. Data Min. Ed. by Wu X, Spiliopoulou M, Wang C, Kumar V, Cao L, Wu Y, Yao Y, and Wu Z. Singapore: *Springer Nature Singapore*, 2025: 95–107.https://doi.org/10.1007/978-981-96-8170-9_8
21. Arcudi A, Frizzo D, Masiero C, Susto GA. Enhancing Interpretability and Generalizability in Extended Isolation Forests. *Eng. Appl. Artif. Intell.* 2024; 138: 109409. https://doi.org/10.1016/j.engappai.2024.109409
22. Cao Y, Xiang H, Zhang H, Zhu Y, Ting KM. Anomaly Detection Based on Isolation Mechanisms: A Survey. *Machine Intelligence Research* 2025; 22: 849–65. https://doi.org/10.1007/s11633-025-1554-4
23. Han S, Hu X, Huang H, Jiang M, Zhao Y. Adbench: Anomaly Detection Benchmark. *Adv. Neural Inf. Process. Syst.* 2022; 35: 32142–59. https://doi.org/10.52202/068431-2329