A Review of COPOD and ABOD: Methodologies and Performance Comparison

MOHAIMINUL ISLAM

Outlier detection involves identifying data points that deviate significantly from the general characteristics of a given data distribution. This report summarizes two parameter-free probabilistic outlier detection methods: COPOD from [6] and ABOD from [5], along with a comparative analysis and discussion of their limitations. COPOD leverages empirical copulas to estimate tail probabilities, while ABOD ranks data points based on the variance of angles between difference vectors. COPOD generally works well in low- and medium-dimensional datasets with simple linear or quadratic relationships. Conversely, ABOD is particularly effective in datasets with more complex geometric patterns, where its angle-based approach can capture subtle variations that signal outliers. While both methods are non-parametric, ABOD comes with higher computational complexity and requires higher time cost. Both methods are optimized for numerical data and perform poorly in data spaces with a high proportion of categorical features.

ACM Reference Format:
Mohaiminul Islam. 2025. A Review of COPOD and ABOD: Methodologies and Performance Comparison. 1, 1 (February 2025), 9 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Outlier detection, a cornerstone of data analysis, aims to identify rare and anomalous observations that deviate significantly from normal data points. These outliers often carry critical insights, signaling errors, fraud, or novel patterns in applications ranging from fraud detection to healthcare diagnostics. Numerous methods have been developed for outlier detection, including statistical approaches, clustering-based techniques, and machine-learning models. However, traditional outlier detection methods—such as distance-based approaches (e.g., k-nearest neighbors) or density-based methods (e.g., Local Outlier Factor, LOF)—struggle with high-dimensional data, where the curse of dimensionality renders distance metrics less meaningful and significantly increases computational costs. This limitation has spurred the development of more advanced algorithms, such as COPOD and ABOD, which are specifically designed to address the challenges posed by high-dimensional datasets.

COPOD introduces a fundamental shift in outlier detection by leveraging copula models to estimate tail probabilities, moving away from traditional distance- and density-based approaches. This paradigm shift enables parameter-free, interpretable, and scalable anomaly detection, making COPOD particularly effective in high-dimensional data.In contrast, ABOD overcomes the pitfalls of traditional distance-based methods by analyzing the variance of angles between data points, offering a geometric perspective that remains robust in high-dimensional spaces. While both algorithms share the goal of overcoming dimensionality-related limitations, they diverge fundamentally in methodology. COPOD focuses on statistical dependencies between variables using empirical copulas, whereas ABOD quantifies outlierness through angular relationships in vector spaces. Despite

Author's address: Mohaiminul Islam, mohaiminul.islam@tu-dortmund.de.

XXXX-XXXX/2025/2-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

their strengths, both methods face unique limitations—such as COPOD's sensitivity to skewed distributions and ABOD's computational complexity—that warrant careful consideration in practical applications.

This report provides a comparative analysis of COPOD and ABOD, focusing on their theoretical foundations, methodological differences, and limitations. By summarizing their underlying principles and evaluating their effectiveness, this study highlights the strengths and weaknesses of each method, offering insights into their comparative performance across different datasets.

2 RELATED WORK

Outlier detection has evolved significantly over the past decades, with methodologies spanning statistical, proximity-based, density-based, and machine-learning approaches. Early techniques, such as statistical methods e.g., Z-score, Grubbs' test. These methods relied on assumptions about data distributions (e.g., Gaussianity) and only examine single attributes to flag deviations [2] from the corresponding central values. These methods were extended to multivariate analysis using Gaussian Mixture Models (GMMs)- where dataset set is assumed to have k Gaussian distribution and for each observation the generative probability under these distributions is estimated via maximum likelihood fit[12]. However, GMMs require strong distributional assumptions and the the corresponding parameters are data-driven, hence the inclusion of outliers in the computational step hinders the robustness of these methods.

Apart from these parametric methods, the depth-based methods -which adopted geometric principles to identify outliers are developed. Inspired by computational geometry, these approaches organized data into layers of convex hulls or depth contours, with outliers occupying the outermost layers with shallow depth [8]. Due to the exponential complexity of constructing convex hulls, these methods require exponential computational cost in high dimensional data space, thus making them infeasible beyond low dimensional settings.

The distance-based methods i.e. k-nearest neighbors addressed this limitation by ranking outliers based on distances to their k nearest neighbors and selecting the top n-ranked objects [10]. However, these methods faltered under the curse of dimensionality, where distances between points became indistinguishable in high-dimensional spaces [3]. The density-based methods like LOF mitigate this limitation which compares the density of each objects with density of it's k neighbor points. While LOF improved robustness in clustered data, it remained computationally intensive and sensitive to parameter choices, such as the number of neighbors k [4].

The development of ensemble methods, such as Isolation Forest, introduced scalable solutions for high-dimensional data which partitioned the feature space recursively to isolate the anomalies with a tree-based approach. While the anomalies got isolated in fewer split compare to the normal points and the anomaly score is determined based on the average path length across all trees. For complex data types like images, text, or sensor signals, deep autoencoders emerged, leveraging neural networks to compress inputs into latent representations and reconstruct them. Observations with high reconstruction errors—indicating the model's inability to reproduce them—are flagged as outliers [15]. Beyond traditional numerical datasets, graph-based techniques extend outlier detection to relational data, such as social networks or biological interaction graphs. These methods identify anomalies through structural irregularities, such as nodes with unexpected connection patterns i.e. fraudsters in financial transactions or edges linking disparate communities [1].

In summary, outlier detection techniques leverage characteristics such as distance, depth, density, and structure to identify anomalies. This report focuses on two novel methods, Copula-Based Outlier Detection (COPOD) and Angle-Based Outlier Detection (ABOD), and explores their underlying intuition for detecting anomalous data points.

3 METHODS

3.1 COPOD-Copula Based Outlier Detection

A **copula** is a mathematical function that captures the dependency structure between random variables, allowing the modeling of their joint distribution independently from their marginal distributions. Formally, a copula is a multivariate cumulative distribution function (CDF) $C : [0,1]^d \to [0,1]$ for a random vector $u = (U_1, U_2, \ldots, U_d)$ with uniform $U(0,1)$ marginals. A copula can be expressed as:

$$C_U(u) = P(U_1 \leq u_1, \ldots, U_d \leq u_d) \tag{1}$$

where $P(U_j \leq u_j) = u_j$ for $j \in \{1, \ldots, d\}$ and $u_j \in [0,1]$. Thus, a copula links the marginal CDFs of individual random variables to their joint CDF.

Sklar's Theorem [13]states that for any random vector $x = (X_1, X_2, \ldots, X_d)$ with joint distribution $F(x_1, x_2, \ldots, x_d)$ and marginals $F_1(x_1), F_2(x_2), \ldots, F_d(x_d)$, it can be expressed as:

$$F(x) = C(F_1(x_1), \ldots, F_d(x_d)) \tag{2}$$

Sklar's theorem also shows that when the marginals are univariate, the equation holds, allowing us to model each dimension separately to capture the joint distribution. This provides significant flexibility in modeling the dependencies across dimensions.

Additionally, a uniform random variable $U(0,1)$ can be transformed into any desired distribution $F(x)$ using its inverse CDF:

$$X_j = F_j^{-1}(U_j) \tag{3}$$

Based on the intuition described above, the **COPOD** algorithm is developed, which utilizes data-driven, non-parametric copula estimation through the empirical copula distribution function (ECDF):

$$F(x) = P(X \leq x) = \frac{1}{n} \sum_{i=1}^{n} I(X_i \leq x)$$

By using the inverse CDF, the marginal distributions of each dimension are transformed into copula observations $U_d \sim U(0,1)$. Therefore:

$$(\hat{U}_1, \ldots, \hat{U}_d) = (\hat{F}_1(X_1), \ldots, \hat{F}_d(X_d))$$

Leveraging the definition of copulas from equation (1) and equation (2), we get:

$$F(x) = C(F_1(x_1), \ldots, F_d(x_d)) = C(u_1, \ldots, u_d)$$

As u_j's are independent and uniform.This is approximated by:

$$\frac{1}{n} \sum_{i=1}^{n} I(\hat{U}_1 \leq u_1, \ldots, \hat{U}_d \leq u_d) = u_1.u_2 \ldots u_d$$

COPOD estimates the extremeness of each observation in the dataset. For a d-variate observation x_i, COPOD estimates the probability of observing a value as extreme as x_i, i.e., $F(x) = P(X \leq x_i)$ or $1 - F_X(x_i) = P(X > x_i)$. Depending on the tail probabilities used, the method is referred to as either left tail or right tail probability.

The **right tail probability** is derived as:

$$C(1-u) = P(U_1 \leq u_1, \ldots, U_d \leq u_d)$$

This is computed by calculating the copula based on F(-x) :

$$1 - F(x) = P(X > x) = P(-X \leq -x) = F(-x)$$

However, as the dimension d increases, the copula function tends to zero. To improve scalability, a **logarithmic transformation** is applied to each copula observation, resulting in an outlier score, a relative measure of how extreme x_i is compared to other points in the dataset. The final equation becomes:

$$-\log(\hat{C}(u)) = -\sum_{j=1}^{d} \log(P(\hat{U}_j \leq u_j)) = -\sum_{j=1}^{d} \log(u_j)$$

Whether to use left or right tail probabilities depends on the **skewness** of the dataset. The skewness of each dimension is computed as:

$$b_j = \frac{1}{n} \sum_{i=1}^{n} (X_{j,i} - \bar{X}_j)^3 \Bigg/ \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (X_{j,i} - \bar{X}_j)^2}$$

for $b = [b_1, \ldots, b_d]$. If $b_d < 0$, the left tail probability is used; otherwise, the right tail probability .

In summary, COPOD first constructs the empirical CDF for each dimension, transforms the marginals into copula observations using the inverse CDF, and applies the copula function with negative log probabilities. For robustness, COPOD calculates outlier scores using left-tailed, right-tailed, and skewness-corrected copula functions. It then considers the maximum of these quantities as the final outlier score for each observation.

3.2 Angle-Based Outlier Detection (ABOD)

Angle-Based Outlier Detection (ABOD) labels a data point as an outlier or inlier by leveraging the angles between difference vectors from a reference point to all other points, rather than relying solely on distances as an identification criterion. Since angular relationships remain stable with increasing dimensionality, ABOD mitigates the limitations of distance-based methods such as k-nearest neighbors (K-NN), which lose discriminative power due to the Curse of Dimensionality.

Formally, given a dataset $D \subset \mathbb{R}^d$, let $A \in D$ be a reference point, and let $B, C \in D \setminus \{A\}$ be two other points. The difference vectors $\overrightarrow{AB}$ and $\overrightarrow{AC}$ are used to compute the Angle-Based Outlier Factor (ABOF) of A, which is defined as the variance of the angles weighted by their respective distances:

$$ABOF(A) = \mathrm{VAR}_{B,C \in D} \left(\frac{\langle \overrightarrow{AB}, \overrightarrow{AC} \rangle}{\|\overrightarrow{AB}\|^2 \cdot \|\overrightarrow{AC}\|^2} \right) \tag{4}$$

where $\langle \cdot, \cdot \rangle$ represents the scalar (dot) product, and the angle θ between the vectors is given by:

$$\cos(\theta) = \frac{\langle \overrightarrow{AB}, \overrightarrow{AC} \rangle}{\|\overrightarrow{AB}\| \cdot \|\overrightarrow{AC}\|} \tag{5}$$

Thus, the ABOF computation accounts for both angular variance and the magnitude of the difference vectors. While distance still influences the outcome, its effect is minor compared to the angular component.

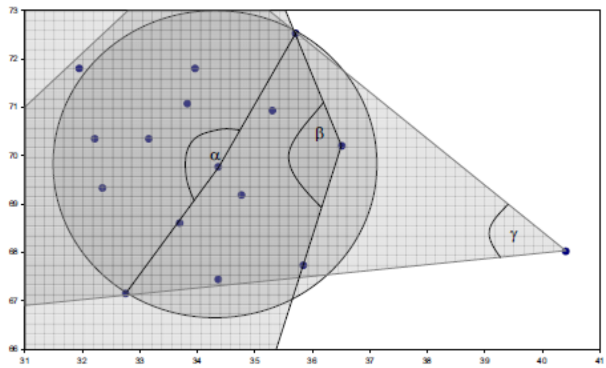

Fig. 1. Intuition of Angle-Based Outlier Detection

The intuition behind ABOD is illustrated in Figure 1 . Inliers, being surrounded by many data points, exhibit widely varying angles between difference vectors to pairs of other points, increasing the variation of angles and resulting in a higher ABOF. Conversely, outliers, which typically reside at the periphery of a cluster, generate a more stable spectrum of angles, leading to a smaller ABOF value.

For each data point, the ABOF value is computed and ranked in ascending order. Points with the lowest ABOF values are detected as potential outliers. However, since ABOF requires calculating angles between all other points for each data point, its time complexity is $O(n^3)$, making it impractical for large-scale datasets. To speed up the process, two approximation algorithms for ABOD have been developed: **Fast ABOD** and **LB-ABOD**.

3.2.1 Fast ABOD. In Fast ABOD, instead of using all data points, the ABOF factor is calculated using only the k nearest neighbors of a reference point. Mathematically, the ABOF computation in equation 4 can be modified as follows:

$$\text{approxABOF}_k(A) = \text{VAR}_{B,C\in N_k(A)}\left(\frac{\langle\overrightarrow{AB},\overrightarrow{AC}\rangle}{\|\overrightarrow{AB}\|^2\cdot\|\overrightarrow{AC}\|^2}\right) \tag{6}$$

Fast ABOD considers only a subset of points with higher weights, increasing the influence of the distance metric. This approximation works well in low-dimensional settings but depends on the number of nearest neighbors (k) and the quality of neighbor selection.

3.2.2 LB-ABOF. :The LB-ABOF is also approximated using k nearest neighbors, but it conservatively estimates a lower bound for the true ABOF. The mathematical equation for LB-ABOF is as follows:

$$LB\text{-}ABOF_k(\vec{A}) = \frac{\sum_{\vec{B}\in N_k(\vec{A})}\sum_{\vec{C}\in N_k(\vec{A})}\left(\frac{1}{\|\vec{AB}\|\cdot\|\vec{AC}\|}\cdot\frac{\langle\vec{AB},\vec{AC}\rangle}{\|\vec{AB}\|^2\cdot\|\vec{AC}\|^2}\right)^2+R_1}{\sum_{\vec{B}\in D}\sum_{\vec{C}\in D}\frac{1}{\|\vec{AB}\|\cdot\|\vec{AC}\|}} - \left(\frac{\sum_{\vec{B}\in N_k(\vec{A})}\sum_{\vec{C}\in N_k(\vec{A})}\frac{1}{\|\vec{AB}\|\cdot\|\vec{AC}\|}\cdot\frac{\langle\vec{AB},\vec{AC}\rangle}{\|\vec{AB}\|^2\cdot\|\vec{AC}\|^2}+R_2}{\sum_{\vec{B}\in D}\sum_{\vec{C}\in D}\frac{1}{\|\vec{AB}\|\cdot\|\vec{AC}\|}}\right)^2$$

Here, R_1 and R_2 account for the influence of points outside the nearest neighbors. To ensure that the condition $\text{ABOF}(A) - \text{LB-ABOF}_k(A) \geq 0$ holds, R_1 needs to be as small as possible, and R_2 needs to be as large as possible. Thus, R_1 is minimized by assuming missing angles are orthogonal ($\cos\theta = 0$), leading to $R_1 = 0$, while R_2 is maximized by assuming missing angles have maximum cosine similarity ($\cos\theta = 1$), resulting in a conservative approximation of ABOF.

In summary, the LB-ABOF algorithm starts by identifying the k-nearest neighbors for each point in the dataset. It computes the LB-ABOF for all points and organizes them in a candidate list sorted by their LB-ABOF values. The algorithm calculates the exact ABOF for the top candidates and adds them to the result list. It iteratively examines the next best candidate, replacing objects in the result list if a smaller ABOF is found until the largest ABOF in the result list is smaller than the smallest approximated ABOF in the candidate list, at which point the process terminates. This ensures that outliers are reliably identified without exhaustive computation.

4 DISCUSSION AND CRITIQUE

4.1 COPOD

As stated above, COPOD offers a non-parametric approach to outlier detection by modeling multivariate dependencies through the empirical copula, which relies on rank-based transformation of features using the empirical cumulative distribution function (ECDF). As COPOD evaluates anomalous behavior on a feature basis, it enables tracking the outlier score for each feature, making it a highly interpretable method. Conversely, because of using ECDF for constructing copula, this method inherently assumes a continuous feature space, making it less effective for modeling dependencies in categorical or mixed-type data. This is because categorical features lack the natural ordering required for meaningful rank transformations. For example, encoding discrete categories like occupation or product type as ordinal or one-hot features introduces artificial sparsity, distorting the copula's dependency structure.

Furthermore, COPOD's tail probability estimates—how extreme a data point is relative to the joint distribution of variables—can be unreliable when strong and complex correlations exist among features. For instance, in a correlated 2D Gaussian distribution with mean $(0, 0)$ and $\rho = 0.5$, the true joint probability of obtaining $(0, 0)$ is 0.33, but COPOD estimates it as 0.25. This discrepancy arises primarily due to its reliance on rank-transformed features. The rank transformation approximates dependencies but fails to fully capture the nuanced correlation structure, thus weakening its ability to accurately model the true joint distribution. As a result, COPOD performs poorly when variables are highly correlated, leading to unreliable probability estimates.

While the empirical copula converges to the true copula asymptotically [9], this demonstrates that copula estimation is not deterministic but is instead sensitive to sample size. Furthermore, since COPOD estimates tail probabilities from the joint distribution, it requires a large number of samples to have enough observations in high-dimensional tail regions. For instance, in a 10-dimensional dataset with only 200 samples, fewer than 0.1% of points occupy the 10-variate tails, rendering COPOD's empirical copula unreliable. Thus, COPOD performs poorly on small datasets, limiting its practical applicability in low-sample scenarios.

4.2 ABOD

As discussed previously, ABOD considers the angular relationships between points, weighted by the distance between them. For each data point, ABOD evaluates all possible pairs of points to calculate these angular relationships. This results in a time complexity of $O(n^3)$, making the algorithm computationally expensive, especially for high-dimensional datasets. Although approximation techniques, such as Fast ABOD using k-nearest neighbors, can help reduce the computational burden, these methods introduce additional challenges. The accuracy and determinism of the results become heavily dependent on the choice of k and the quality of the selected neighbors. Consequently, the approximated results may vary, reducing the reliability of outlier detection.

Another fundamental limitation is that ABOD only considers the relationship between each point and its neighbors, but does not account for the density or distribution of the neighborhood. For instance, consider Figure 2 ,

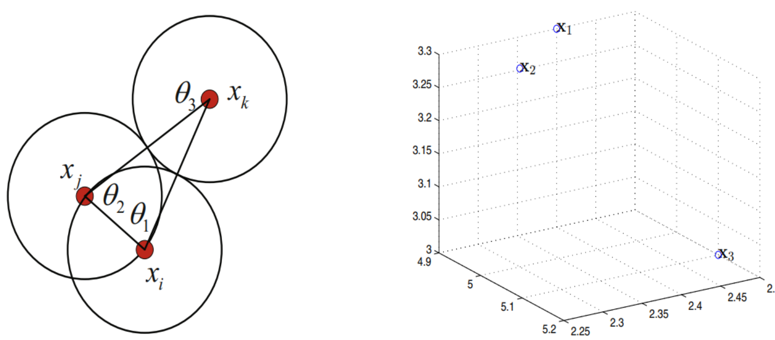

Fig. 2. Illustration of the ABOD outlier detection scenario. The point X_k is spatially distant from the cluster but is angularly aligned with X_i and X_j.

where we assume $\theta_1 = \theta_2 > \theta_3$ and $\|x_j x_k\| = \|x_i x_k\| > \|x_i x_j\|$. In this case, X_k is expected to be an outlier in the plot and should thus have the lowest ABOF value, i.e., $ABOF(x_k) < ABOF(x_i) = ABOF(x_j)$. However, as shown in [7], when the following condition holds:

$$\|x_i x_j\| < \|x_k x_i\| < \frac{\cos(\theta_3)}{\cos(\theta_1)}\|x_i x_j\| = \frac{\cos(\theta_3)}{\cos(\theta_2)}\|x_j x_i\|$$

ABOD incorrectly assigns the highest ABOF value to X_k, due to the dominance of $\cos(\theta_3)$. In this case, although X_k is spatially distant from the cluster, its angular alignment with its neighbors causes ABOD to mistakenly classify it as an inlier. This demonstrates a fundamental weakness in ABOD's assumption that outliers disrupt angular geometry. Spatially distant outliers can still maintain angular alignments with clusters, especially if they reside in sparsely populated regions. In such cases, distance-based methods outperform ABOD, as they better account for the spatial distribution of points.

5 PERFORMANCE COMPARISON: ABOD & COPOD

The performance of 10 outlier detection algorithms is compared in [6], using the PYOD [14] library on 30 benchmark data sets from the ODDS repository [11]. For this report, only the performance of ABOD and COPOD has been extracted in table 1 from where it is observed that COPOD outperforms ABOD in terms of ROC-AUC in 24 out of 30 datasets, while ABOD achieves higher ROC-AUC in 6 datasets. Furthermore, when considering the precision metric, COPOD achieves a higher precision of 0.56, compared to 0.35 for ABOD, demonstrating its superior overall performance. These results underscore that the performance of ABOD and COPOD is highly dataset-dependent. While COPOD tends to excel in most datasets due to its ability to model relationships through copulas, ABOD is effective in specific datasets where its geometric approach is more suited to the data structure. Therefore, understanding the nature of the dataset and its feature dependencies is crucial when selecting the appropriate outlier detection method.

6 CONCLUSION

In this report, the two outlier detection algorithms, COPOD and ABOD, are presented with a brief description of their methodology and workflow, while also highlighting the limitations each algorithm possesses. Additionally, a comparative analysis using AUC-ROC and precision as evaluation metrics is presented, which demonstrates that while each method employs its own generalization technique to model relationships, the efficiency of the model depends on the characteristics of the dataset. Hence, selecting an appropriate algorithm requires understanding the underlying data distribution and structure.

REFERENCES

[1] Leman Akoglu, Hanghang Tong, and Danai Koutra. 2015. Graph-Based Anomaly Detection and Description: A Survey. *Comput. Surveys* 48, 1 (2015), 1–38.
[2] Vic Barnett and Toby Lewis. 1994. *Outliers in Statistical Data* (3 ed.). Wiley, New York.
[3] Kevin Beyer, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft. 1999. When is "Nearest Neighbor" Meaningful?. In *Proceedings of the 7th International Conference on Database Theory (ICDT)*. 217–235.
[4] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. 2000. LOF: Identifying Density-Based Local Outliers. In *ACM SIGMOD Record*, Vol. 29. 93–104.
[5] Hans-Peter Kriegel, Matthias Schubert, and Arthur Zimek. 2008. Angle-Based Outlier Detection in High-Dimensional Data. *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)* 14 (2008).
[6] Xin Li et al. 2020. COPOD: A Copula-based Outlier Detection Algorithm. *Journal Name* 34 (2020), 123–145.
[7] Xiaojie Li, Jian Cheng Lv, and Dongdong Cheng. 2020. Angle-Based Outlier Detection Algorithm with More Stable Relationships. *Machine Intelligence Laboratory, College of Computer Science, Sichuan University* (2020).
[8] Regina Y. Liu, Jesse M. Parelius, and Kesar Singh. 1999. Multivariate Analysis by Data Depth: Descriptive Statistics, Graphics and Inference. *Annals of Statistics* 27, 3 (1999), 783–858.
[9] R. B. Nelsen. 2007. *An Introduction to Copulas*. Springer Science, New York.
[10] Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim. 2000. Efficient Algorithms for Mining Outliers from Large Data Sets. In *ACM SIGMOD Record*, Vol. 29. 427–438.
[11] S. Rayana. 2016. ODDS Library: Outlier Detection DataSets. https://paperswithcode.com/dataset/odds Accessed: 2025-02-23.
[12] Douglas A. Reynolds. 2009. Gaussian Mixture Models. In *Encyclopedia of Biometrics*, Stan Z. Li and Anil K. Jain (Eds.). Springer, Boston, MA, 659–663.
[13] A. Sklar. 1959. Fonctions de répartition à n dimensions et leurs marges. *Publications de l'Institut de Statistique de l'Université de Paris* 8 (1959), 229–231.
[14] Yue Zhao, Zheng Li, Chi Zhang, Cezar Ionescu, and Xiyang Hu. 2019. PYOD: A Python Toolbox for Scalable Outlier Detection. *Journal of Machine Learning Research* 20 (2019), 1–7. https://www.jmlr.org/papers/volume20/19-335/19-335.pdf
[15] Chong Zhou and Randy C. Paffenroth. 2017. Anomaly Detection with Robust Deep Autoencoders. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. ACM, 665–674.

A ADDITIONAL TABLES

Data	**Avg ROC-AUC**		**Average Precision**	
	ABOD	**COPOD**	**ABOD**	**COPOD**
Arrhythmia (mat)	0.7688	**0.8021**	0.3585	**0.4727**
Breastw (mat)	0.3959	**0.9936**	0.2945	**0.9877**
Cardio (mat)	0.5692	**0.8974**	0.1936	**0.5793**
Ionosphere (mat)	**0.9248**	0.8307	**0.9141**	0.7194
Lympho (mat)	0.911	**0.9935**	0.5168	**0.8936**
Mammography (mat)	0.5494	**0.8942**	0.0232	**0.429**
Optdigits (mat)	0.4667	**0.7348**	0.0279	**0.0532**
Pima (mat)	**0.6794**	0.6638	0.5107	**0.5407**
Satellite (mat)	0.5714	**0.6616**	0.3973	**0.8599**
Satimage-2 (mat)	0.819	**0.9852**	0.1874	**0.9015**
Shuttle (mat)	0.6517	**0.9982**	0.1708	**0.9873**
Speech (mat)	**0.6267**	0.4845	**0.0395**	0.0195
WBC (mat)	0.9047	**0.9747**	0.3545	**0.556**
Wine (mat)	0.3405	**0.9949**	0.0838	**0.682**
Arrhythmia (arff)	0.7396	**0.7618**	**0.6987**	0.4046
Cardioto (arff)	0.5493	**0.8347**	0.247	**0.3778**
HeartDisease (arff)	0.5969	**0.6728**	0.5341	**0.6149**
Hepatitis (arff)	0.7155	**0.8438**	0.3304	**0.5849**
InternetAds (arff)	0.6427	**0.6733**	0.276	**0.502**
Ionosphere (arff)	**0.9250**	0.8219	**0.5055**	0.4637
KDDCup99 (arff)	0.6925	**0.9985**	0.3709	**0.7031**
Lymphography (arff)	0.9864	**0.9985**	0.2465	**0.8812**
Pima (arff)	0.6665	**0.6845**	**0.9145**	0.6132
Shuttle (arff)	**0.8338**	0.8795	0.0182	**0.2389**
SpamBase (arff)	0.4349	**0.7388**	**0.8018**	0.3945
Stamps (arff)	0.7617	**0.8832**	0.2625	**0.3882**
Waveform (arff)	0.6522	**0.6648**	0.0656	**0.0855**
WBC (arff)	**0.9559**	0.9477	0.5422	**0.7085**
WDBC (arff)	0.9032	**0.9545**	0.43	**0.8407**
WPBC (arff)	0.457	**0.5465**	0.2095	**0.2438**
AVG	0.6900	**0.8247**	0.3512	**0.5649**

Table 1. Performance comparison of ABOD and COPOD on various datasets