A REVEALING LARGE-SCALE EVALUATION OF UNSUPERVISED ANOMALY DETECTION ALGORITHMS

Maxime Alvarez*, **Jean-Charles Verdier***, **D'Jeff K. Nkashama***
Marc Frappier, Pierre-Martin Tardif, Froduald Kabanza

GRIC, Université de Sherbrooke
Sherbrooke, QC, Canada
{maxime.alvarez,jean-charles.verdier,djeff.nkashama.kanda
marc.frappier,pierre-martin.tardif,froduald.kabanza}@usherbrooke.ca

ABSTRACT

Anomaly detection has many applications ranging from bank-fraud detection and cyber-threat detection to equipment maintenance and health monitoring. However, choosing a suitable algorithm for a given application remains a challenging design decision, often informed by the literature on anomaly detection algorithms. We extensively reviewed twelve of the most popular unsupervised anomaly detection methods. We observed that, so far, they have been compared using inconsistent protocols – the choice of the class of interest or the positive class, the split of training and test data, and the choice of hyperparameters – leading to ambiguous evaluations. This observation led us to define a coherent evaluation protocol which we then used to produce an updated and more precise picture of the relative performance of the twelve methods on five widely used tabular datasets. While our evaluation cannot pinpoint a method that outperforms all the others on all datasets, it identifies those that stand out and revise misconceived knowledge about their relative performances.

1 INTRODUCTION

A tenet of scientific publications is that the published results should be fair, unambiguous, and reproducible. However, there is an increasing awareness that this is not always the case in published machine learning research Gorman & Bedrick (2019); Agarwal et al. (2021); Kadlec et al. (2017); Fourure et al. (2021); Musgrave et al. (2020); Marie et al. (2021); Raff (2019). To illustrate, sometimes methods are compared while using inconsistent hyparameter tunings Kadlec et al. (2017); Musgrave et al. (2020) or misleading metrics Musgrave et al. (2020), resulting in unfair comparative evaluations. Different choices for the anomaly class, differences in training protocol settings, and different proportions of anomalies in the test set were also identified as leading to inconsistent evaluations of machine learning models Fourure et al. (2021). Other studies, in deep reinforcement learning for example Agarwal et al. (2021), stress the importance of considering the statistical uncertainty uncured by comparing machine learning models on a small number of training runs, to ensure reliable performance evaluations.

This paper analyses twelve of the most popular unsupervised anomaly detection methods. We show that some of the issues raised by the abovementioned work also apply to the anomaly detection literature specifically. Anomaly detection algorithms attracted our attention because they are essential to a wide range of applications, ranging from bank-fraud detection Zhu et al. (2021) and cyber-threat detection Tan et al. (2011); Schubert et al. (2014) to equipment maintenance Carvalho et al. (2019) and health monitoring Wei et al. (2018).

An anomaly is an observation that deviates from what is deemed normal observations Ruff et al. (2021). Anomaly detection algorithms are by nature classification algorithms and, like classification methods in general, are grouped into parametric and nonparametric approaches Ruff et al. (2021);

[*] Equal contribution. Corresponding author: maxime.alvarez@usherbrooke.ca

Kwon et al. (2019); Chalapathy & Chawla (2019). Parametric approaches assume a model defined by some parameters. Examples include decision-boundary learning methods, such as OC-SVM Schölkopf et al. (1999) and DeepSVDD Ruff et al. (2018), which learn a separation between normal and abnormal samples; reconstruction methods, such as MemAE Gong et al. (2019) and DAE Chen et al. (2018), which learn to reconstruct normal samples and classify samples that cannot be properly reconstructed as anomalous; and probabilistic methods, like DAGMM Zong et al. (2018), which learn the probability density function of the normal data. Nonparametric methods do not assume any parametric model. They include distance-based methods, like LOF Breunig et al. (2000) and RecForest Xu et al. (2021), which learn to identify anomalies by their distance to normal samples, given some distance-based metric.

The twelve anomaly detection methods we analyzed are: Deep Auto-Encoder Chen et al. (2018), NeuTraLAD Qiu et al. (2021), DAGMM Zong et al. (2018), SOM-DAGMM Chen et al. (2021), DUAD Li et al. (2021), MemAE Gong et al. (2019), ALAD Zenati et al. (2018), DSEBM Zhai et al. (2016), DROCC Goyal et al. (2020), DeepSVDD Ruff et al. (2018), LOF Breunig et al. (2000) and OC-SVM Schölkopf et al. (1999). After pointing to inconsistencies in published results for these methods, we propose a rigorous evaluation protocol and apply it to reevaluate those methods on five widely used tabular datasets: KDDCUP, NSL-KDD, CSE-CIC-IDS2018, Arrhythmia, and Thyroid. The new evaluation shows that some methods, previously shown to perform better than others, do not perform as well under our proposed protocol. We hope that the updated evaluation of anomaly detection algorithms will better inform future design choices of anomaly detection methods and future baselines for new method discoveries. We also believe that the adoption by authors of a coherent evaluation protocol like the one we propose will contribute to remedying the increasing concern of unfair, ambiguous, and difficult-to-reproduce experimental results in the machine learning literature.

2 Issues with existing evaluations of machine-learning-based anomaly detection algorithms

While reviewing the literature on machine-learning-based anomaly detection algorithms, we noticed inconsistencies in protocols used to evaluate and compare different algorithms, especially for the splitting between training and test datasets, the choice of performance metrics, and the threshold used to flag anomalies. We also observed ambiguity in the definition of the positive class (i.e., the class of interest) used for different models evaluations. As a result of these inconsistencies, it is difficult to make sense of the experimental evaluations from one paper to another.

Data splitting. Different train-test data splits have been used while comparing the performances of different algorithms. One must usually decide on the training and test sets splitting. In addition, for anomaly detection, one must decide whether any of these two sets will contain normal data, abnormal data, or both. Whatever the decision, it should be made consistently when evaluating different algorithms. Otherwise, the comparisons of one paper to another are meaningless. It turns out that, in the literature, the data split decisions are indeed inconsistent. As an example, to compare different algorithms, some approaches split the data according to the following strategy Zong et al. (2018); Zenati et al. (2018); Bergman & Hoshen (2020): the training dataset consists of 50% of the normal data, whereas the test set consists of the remaining 50% normal data plus anomalies. Based on this split, conclusions are made on the relative performance of different algorithms in the literature. We call this the “Recycling strategy”.

In contrast, Zhai et al. (2016) start with a set containing both the normal data and anomalies, splits it evenly into two sets, then trains on a set that is one of the sets stripped of the anomalies, and tests it on the other set. We refer to this as the “Discarding strategy”. At first glance, the Recycling strategy and the Discarding strategy are similar in that training is done on normal data and testing is done on a mix of normal and abnormal data. The subtle difference here is that now the test set contains only half of the anomalies available in the original data. Considering that anomalies are rare – they constitute a small percentage of the original dataset – that subtle difference could have a significant impact on the testing performance Fourure et al. (2021). Put another way, the two strategies could lead to different conclusions when comparing anomaly detection algorithms on a dataset. For instance, Zenati et al. (2018) refer to measurements made by previous authors using the Discarding strategy to compare against measurements made using the Recycling strategy in their paper. We later demonstrate that this is misleading, as one might expect. Regardless of the inconsistent comparison, since anomalous data is typically scarce, we argue that it should always be injected into the test set to evaluate the capacity of the model in detecting more anomalous signals.

The two data strategies above say nothing about the proportion of anomalous and normal data in the test sets. Some authors fix that proportion to 50% – making the test set balanced, thereby introducing a different test strategy Goyal et al. (2020). We call it the "Balanced test set strategy". On the other hand, while the three strategies discussed so far train only on normal data, methods have also been proposed that train only on abnormal data. As an example, Chen et al. (2021) use 50% of the anomalies as the training set and the rest of the anomaly data plus normal data as the test set.

Performance metrics and threshold. Since most of the datasets in anomaly detection are imbalanced, precision, recall, and F1-score are commonly used metrics to measure model performance and benchmark models. These metrics are computed for a specific threshold in the anomaly detection task – different thresholds may yield different metric values. The work in Fourure et al. (2021) demonstrates that the F1-score with a fixed threshold can be artificially manipulated by increasing or decreasing the number of positive samples in the test set. Therefore, different anomaly ratios in the test set lead to biased comparisons. Moreover, the threshold can be another factor for manipulating performance measures. For instance, Zong et al. (2018), Zenati et al. (2018) and Bergman & Hoshen (2020) set the threshold such that it returns the α^{th} percentile of anomaly scores, with α being the ratio of normal data in the test set; whereas others, like Qiu et al. (2021), search for an optimal threshold, which results in the best performance the model could achieve. Evidently, using different thresholds yields different results and performance comparison is only fair if the models are evaluated using their respective optimal thresholds.

Class of interest (positive class). The choice of the positive class constitutes another source of ambiguity. For instance, Zong et al. (2018) assign the positive class to the minority class (usually anomalous data) whereas, in Goyal et al. (2020) and Chen et al. (2021), the positive class denotes the majority class (normal data). However, with unbalanced data, the F1-score varies for class swapping Chicco & Jurman (2020). Therefore, evaluating the performance of a model on the majority class gives it a clear advantage over those evaluated on the minority class.

Implementation details. Reproducibility in research allows double-checking findings and verifying whether they are reliable. It also facilitates the integration of recent findings when constructing new models. That said, reproducibility remains a challenge in the machine learning community, often due to important missing details in the description of models or the training procedure Pineau et al. (2021). In our literature review on anomaly detection algorithms, we noted similar issues. For instance, Qiu et al. (2021) do not normalize values of features for some datasets, while Zenati et al. (2018), Bergman & Hoshen (2020), Zong et al. (2018) normalize values of attributes for all the datasets. The lack of implementation details may engender serious hurdles in the advancement of research in machine learning, in general; it reduces chances to reproduce results with sufficient certainty and impedes effective and consistent performance comparisons between different models.

3 Proposed training and evaluation protocol

In this section, we propose solutions for the issues identified in the previous section to ensure a fair, reliable, and consistent evaluation, and comparison of anomaly detection algorithms.

3.1 Data split

We propose to partition normal samples into a training and a test set following a 50-50 split using random subsampling. It could also be fair to use another split ratio, as long as all the anomalies are found exclusively within the test set. Given that anomalies are rare by nature and greatly influence the performance metrics, they should be included only in the test set. Also, training on normal data translates seamlessly to real-life applications where most of the data is assumed to be normal. However, for ablation studies, it can be informative to insert a small portion of the abnormal samples during training to study the algorithm's sensitivity to corruption, i.e. how the presence of anomalies in its training data affects its performance.

During our experiments, we trained and tested the models multiple times each. We kept the train-test constant across all runs however it has been pointed out by Bouthillier et al. (2021) that varying sources of randomness give a better estimation of the performances. Thus, it would be recommendable to shuffle all the data before splitting it into the training and test set at the beginning of each run.

3.2 Class of interest

In classification tasks, the class of interest, also called the positive class, is used as the basis for evaluation. Given the large class imbalance in anomaly detection, our protocol defines the minority class as the class of interest. By contrast, using the majority class gives overly optimistic scores and masks the poor performance on the minority class. In most cases, the minority class corresponds to the anomalies.

3.3 Metrics

We propose to use the following metrics to evaluate anomaly detection performance: F1-score, precision, recall, and area under the precision-recall curve (AUPR). AUPR was not used in any of the analyzed papers. As previously mentioned, research shows how F1-score and AUPR are sensitive to class imbalance Jeni et al. (2013); Tharwat (2020) and can be manipulated by changing the anomaly ratio in the test set Fourure et al. (2021). Our protocol mitigates this issue by considering all the anomalies during testing. Thus, the anomaly ratio remains the same for all algorithms. While AUROC is unaffected by skewness in the class distribution, it provides an optimistic view by giving equal weights to predictions on both classes. The AUPR is more sensitive to predictions on the positive class (the anomalies), making it more informative for anomaly detection Saito & Rehmsmeier (2015). Also, AUPR and AUROC are not dependent on the choice of a specific threshold that can prevent comparability Fourure et al. (2021).

3.4 Threshold

A threshold τ must be set to identify anomalies and compute the performance metrics such as F1-score, precision, and recall. Given a ratio of ρ anomalous samples in the test set and the array S of generated scores on the entire test set, an intuitive strategy is to set τ at the $(1-\rho)^{th}$ percentile of scores S. We expect the lowest (or highest, depending on the meaning of the score) of $(1-\rho)^{th}$ percentile to contain the anomalies because they should generate the lowest (or highest scores). Another strategy is to find the optimal threshold, that is, the threshold that maximizes the F1-score. Such a threshold is typically located in the neighbourhood of the $(1-\rho)^{th}$ percentile of scores S. We recommend the use of optimal thresholding for each model for a fair comparison.

As mentioned in the previous section, AUPR and AUROC are not dependent on the choice of a specific threshold. When comparing methods to apply to a problem where the anomaly ratio is unknown, these metrics are more informative than guessing the correct threshold. As such, these metrics are more attractive to industry practitioners.

4 Experiments

This section presents the datasets and the models used in our experiments along with implementation details. We then discuss the results obtained using the evaluation protocol suggested in the previous section.

4.1 Datasets

Commonly used datasets in unsupervised anomaly detection are considered for this task. Table 1 summarizes the information of the different datasets.

- **KDDCUP** is a network intrusion detection dataset widely used as a benchmark in the literature. We use the 10 percent version which uses only 10 percent of the original KDDCUP dataset. It contains 34 continuous and 7 categorical variables that are one-hot encoded. The four different attack scenarios (DOS, R2L, U2R, and probing) are combined

DATASET	NUMBER OF SAMPLES (N)	NUMBER OF FEATURES (D)	ANOMALY RATIO (ρ)
ARRHYTHMIA	452	274	0.1460
CSE-CIC-IDS2018	16 232 944	83	0.1693
KDD 10%	494 021	42	0.1969
NSL-KDD	148 517	42	0.4811
THYROID	3772	6	0.0246

Table 1: General information on the datasets.

into a single "attack" class. After manipulations, we drop the `num_outbound_cmds` and `is_host_login` columns because they both have a single value.

- **NSL-KDD**, provided by the Canadian Institute of Cybersecurity (CIC) (CIC, b), attempts to solve the inherent statistical flaws in KDDCUP (see Tavallaee et al. (2009) for more details) by removing most of the duplicate entries. The result is a much smaller training set with the same variables. The preprocessing steps is identical to KDDCUP. However, the column `is_host_login` is kept because it contains more than one value.
- **CIC-CSE-IDS2018**. This dataset is also provided by CIC. It simulates a complex enterprise network through virtual machines subject to seven different attack scenarios, namely Brute-force, Heartbleed, Botnet, DoS, DDoS, Web attacks, and infiltration of the network from inside (CIC, a). Data cleaning for this dataset replicated the methodology described in Leevy et al. (2021). Attacks are once again combined into a single class.
- **Thyroid**. This classification dataset taken from the ODDS repository (ODDS, b) has three classes but, for the outlier detection task, only the hyperfunction class is treated as the outlier. The other two classes are treated as normal. The attributes are homogeneous and there is no missing or invalid data in this dataset.
- **Arrhythmia**. This multi-class classification dataset, also obtained from the ODDS repository (ODDS, a), combines multiple classes (3, 4, 5, 7, 8, 9, 14, and 15) to form the outlier class while the remaining classes are considered as the normal class. Like Thyroid, our data cleaning pipeline didn't modify the original data.

Min-Max scaling was applied to all the features of the aforementioned datasets. Applying Min-Max scaling on one-hot encoded features has no effect.

4.2 MODELS

Our study compares the performance of 9 recent deep unsupervised learning algorithms tailored for anomaly detection, namely: DAGMM Zong et al. (2018), ALAD Zenati et al. (2018), MemAE Gong et al. (2019), DSEBM Zhai et al. (2016) represented with its two alternative versions DBSEM-e and DBSEM-r, DROCC Goyal et al. (2020), DeepSVDD Ruff et al. (2018), SOM-DAGMM Chen et al. (2021), DUAD Li et al. (2021) and NeuTraLAD Qiu et al. (2021). They were chosen based on their performances and the diversity of approaches. We complement our comparison with two popular baseline methods: OC-SVM Schölkopf et al. (1999), LOF Breunig et al. (2000) and a vanilla Deep Auto Encoder (DAE) Zhou & Paffenroth (2017).

Deep learning methods are implemented using PyTorch and optimized by the Adam algorithm with a learning rate of $1e{-4}$, and training consists of 20 runs. Mini-batch sizes for Arrhythmia, Thyroid, KDD, NSL-KDD, and CSE-CIC-IDS2018 are set to 128, 128, 1024, 1024, and 1024 respectively. The author's PyTorch versions of DeepSVDD, DROCC, and MemAE are integrated into our codebase. For ALAD, we had to convert the original TensorFlow codebase into PyTorch. Scikit-Learn's LOF and OC-SVM implementations are used Pedregosa et al. (2011). Finally, the remaining algorithms (DAGMM, DUAD, SOM-DAGMM, DSEBM, NeuTraLAD) are completely reimplemented because no public repository was made available by the authors. Our code is available on Github[1].

[1] ireydiak/anomaly_detection_NRCAN (github.com)

	KDDCUP 10			NSL-KDD			CSE-CIC-IDS2018		
	Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall	F_1
ALAD	95.1±0.5	96.6±1.0	95.9±0.7	93.6±1.0	90.7±1.9	92.1±1.5	58.9±0.7	59.2±0.2	59.0±0.0
DAE	93.2±1.3	93.2±2.6	93.2±2.0	**97.0±0.1**	95.3±0.2	**96.1±0.1**	67.9±0.4	75.6±0.7	71.5±0.5
DAGMM	93.6±0.9	98.4±1.9	95.9±1.4	89.3±5.5	81.8±9.0	85.3±7.4	48.4±4.1	65.9±7.3	55.8±5.3
DeepSVDD	90.8±2.0	87.6±2.0	89.1±2.0	89.4±2.0	89.2±2.0	89.3±2.0	20.7±11	20.8±11	20.8±11
DROCC	84.0±0.0	99.6±0.0	91.1±0.0	90.4±0.0	90.5±0.0	90.4±0.0	29.6±0.0	**99.6±0.0**	45.6±0.0
DSEBM-e	95.7±0.1	97.6±0.1	96.6±0.1	95.5±0.1	93.7±0.1	94.6±0.1	45.1±0.7	42.7±0.8	43.9±0.8
DSEBM-r	**96.6±0.1**	99.4±0.1	**98.0±0.1**	96.2±.0.1	94.9±0.1	95.5±0.1	42.2±0.1	39.3±0.1	40.7±0.1
DUAD	94.0±0.7	99.1±1.4	96.5±1.0	96.0±0.1	93.2±0.3	94.5±0.2	68.1±3.5	75.8±2.4	**71.8±2.7**
MemAE	93.0±1.2	97.1±2.2	95.0±1.7	96.0±0.0	95.1±0.1	95.6±0.0	60.8±0.1	59.0±0.2	59.9±0.1
NeuTraLAD	93.1±0.3	**99.7±0.1**	96.4±0.2	96.5±0.4	**95.6±0.2**	96.0±0.1	54.6±8.2	65.6±9.8	59.5±8.9
SOM-DAGMM	95.7±0.7	99.8±0.2	97.7±0.3	94.4±1.0	96.8±0.8	95.6±0.3	48.6±1.2	40.3±0.9	44.1±1.1
LOF	93.0±0.0	97.2±0.0	95.1±0.0	88.6±0.0	93.6±0.0	91.1±0.0	75.6±0.0	55.1±0.0	63.8±0.0
OC-SVM	94.2±0.0	99.4±0.0	96.7±0.0	91.5±0.0	94.5±0.0	93.0±0.0	**92.5±0.0**	30.6±0.0	45.4±0.0

(a) Performance metrics on cybersecurity datasets.

	Arrhythmia			Thyroid		
	Precision	Recall	F_1	Precision	Recall	F_1
ALAD	59.5±0.1	55.5±0.8	57.4±0.4	61.5±0.9	77.4±0.8	68.6±0.5
DAE	62.1±2.0	60.9±2.2	61.5±2.5	54.4±4.4	65.5±5.7	59.0±1.5
DAGMM	51.4±3.9	50.0±6.1	50.6±4.7	51.3±7.2	46.3±8.8	48.6±8.0
DeepSVDD	56.4±2.0	54.7±3.0	55.5±3.0	13.0±13	24.7±30	13.1±13
DROCC	26.3±3.6	61.8±15	35.8±2.6	51.8±9.0	77.6±11	62.1±10
DSEBM-e	61.9±1.0	58.1±1.6	59.9±1.0	26.0±0.9	22.0±0.6	23.8±0.7
DSEBM-r	61.8±1.1	58.4±1.3	60.1±1.0	25.8±0.3	21.8±0.4	23.6±0.4
DUAD	58.6±0.4	63.2±1.2	60.8±0.4	12.3±2.0	19.1±1.6	14.9±5.5
MemAE	**63.1±2.1**	62.1±1.5	62.6±1.6	53.4±0.5	59.1±0.5	56.1±0.9
NeuTraLAD	58.5±5.5	63.6±5.3	60.7±3.7	68.9±0.7	**78.5±0.5**	**73.4±0.6**
SOM-DAGMM	51.0±5.9	53.2±7.5	51.9±5.9	61.2±11	49.0±14	52.7±12
LOF	57.1±0.0	66.7±0.0	61.5±0.0	63.0±0.0	75.2±0.0	68.6±0.0
OC-SVM	57.3±0.0	**71.2±0.0**	**63.5±0.0**	**69.6±0.0**	66.6±0.0	68.1±0.0

(b) Performance metrics on medical datasets.

Table 2: Average Precision, Recall, and F1-Score (all with standard deviation) of the twelves models trained exclusively on the normal data.

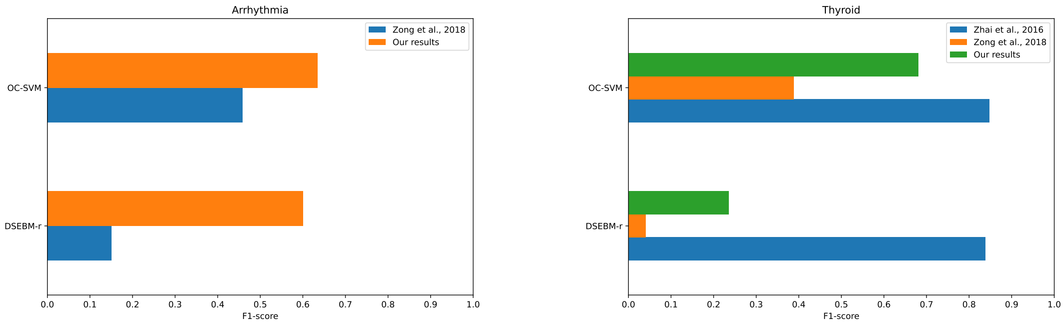
	KDD10		NSL-KDD		CSE-CIC-IDS2018	
	AUROC	AUPR	AUROC	AUPR	AUROC	AUPR
ALAD	99.0±0.2	95.3±1.1	93.9±1.8	94.8±1.7	85.6±2.1	61.5±5.8
DAE	98.2±0.0	94.7±0.1	98.5±0.2	**99.2±0.1**	**88.2±1.8**	68.9±2.5
DeepSVDD	**99.4±0.2**	97.1±1.0	93.1±3.0	95.3±1.0	62.0±7.4	24.2±8.0
DROCC	97.5±0.0	93.2±0.0	94.4±0.0	97.2±0.0	51.1±0.0	29.6±0.0
DSEBM-e	98.6±0.1	93.9±0.5	98.0±0.0	99.0±0.0	70.7±0.2	38.1±0.1
DSEBM-r	99.0±0.0	95.6±0.2	98.3±0.0	99.1±0.0	72.7±1.7	39.8±1.3
DAGMM	99.1±0.3	**97.3±0.6**	94.0±3.6	96.5±2.0	66.7±8.3	50.8±9.5
DUAD	98.3±1.0	93.2±3.5	97.4±0.1	98.6±0.1	82.9±2.3	53.0±0.1
MemAE	98.2±0.2	94.7±0.6	97.9±0.1	98.9±0.1	65.8±2.0	56.6±2.0
NeuTraLAD	98.8±0.1	97.0±0.1	**98.7±0.1**	**99.2±0.1**	81.5±5.6	59.6±6.9
SOM-DAGMM	98.9±0.2	95.8±1.3	98.6±0.2	99.2±0.1	67.9±6.4	53.2±2.4
LOF	91.1±0.0	89.9±0.0	91.1±0.0	89.9±0.0	83.4±0.0	**72.7±0.0**
OC-SVM	98.8±0.0	94.9±0.0	96.5±0.0	96.4±0.0	64.6±0.0	48.2±0.0

(a) AUROC and AUPR scores on cybersecurity datasets.

	Arrhythmia		Thyroid	
	AUROC	AUPR	AUROC	AUPR
ALAD	78.7±1.3	62.0±1.6	85.7±3.9	59.9±4.8
DAE	**81.7±0.6**	**67.5±0.9**	95.1±0.8	54.2±3.1
DAGMM	68.9±2.9	45.8±5.2	84.3±2.6	37.8±5.9
SOM-DAGMM	70.3±5.0	48.7±6.9	85.0±7.1	46.7±13
DUAD	81.2±0.4	66.8±0.4	43.0±0.3	4.6±0.5
MemAE	80.9±0.1	**67.5±0.1**	89.8±4.6	32.7±7.0
DeepSVDD	79.4±0.8	62.5±0.6	83.7±14	51.6±18
DROCC	64.0±4.3	30.8±3.9	95.6±3.8	68.9±15
DSEBM-e	80.1±0.2	67.0±1.0	85.1±0.2	24.3±0.6
DSEBM-r	80.4±2.2	66.9±2.0	85.6±0.1	25.0±0.6
NeuTraLAD	78.9±2.6	63.4±3.3	**98.2±2.2**	**73.9±2.9**
LOF	81.3±0.0	67.0±0.0	97.2±0.0	72.2±0.0
OC-SVM	80.0±0.0	64.0±0.0	96.9±0.0	61.4±0.0

(b) AUROC and AUPR on medical datasets

Table 3: Average AUROC and AUPR (all with standard deviation) of the twelves models trained exclusively on the normal data.



(a) On Arrhythmia.

(b) On Thyroid.

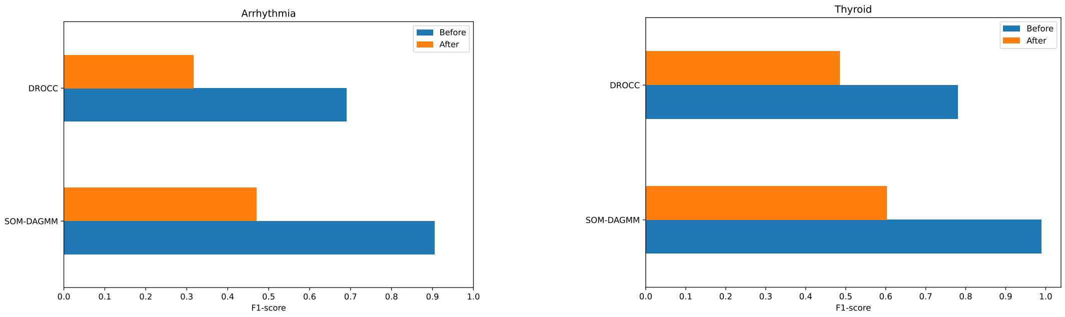
Figure 1: Reported F1-scores for OC-SVM and DSEBM-r by different authors.

4.3 Results

We now highlight the different results using our experimental protocol. Recall, precision, and F1-score are presented in Table 2. Table 3 displays the AUPR results.

Important differences with reported results. We first note significant differences in reported precision, recall, and F1-scores between a few models. Zong et al. (2018) report an F1-score of 0.0403 and 0.1510 for DSEBM-r on Arrhythmia and Thyroid respectively, while the original authors of the method Zhai et al. (2016) obtained 0.8386 on Thyroid. Our results are quite different from both papers: 0.601 and 0.236 on Arrhythmia and Thyroid respectively. Results on baseline models also vary greatly. Figures 1a and 1b summarize these discrepancies. It's unclear whether the experiments integrated all anomalies during testing or if some of them were left out during the subsampling phase. This would explain the large differences. Zenati et al. (2018) and Gong et al. (2019) cite the results from Zong et al. (2018) on these models. Interestingly, the DSEBM implementation described in Zenati et al. (2018), available on Github[2], generates results very different from those reported in Zenati et al. (2018). They are, in fact, more in line with our scores.

Taking the majority class as the class of interest yields overly optimistic results. As displayed in Figures 2a and 2b, results on DROCC and SOM-DAGMM differ significantly when considering the minority class as the class of interest. Results drop from 0.78, 0.69 to 0.485 (-0.295) and 0.317 (-0.3729) on Thyroid and Arrhythmia respectively for DROCC. Similarly, our protocol generates 0.471 and 0.602 on the same datasets compared to 0.9053 (-0.4343) and 0.9888 (-0.3868) reported by SOM-DAGMM. Emphasizing predictions on the normal class can be misleading when dealing with skewness in class distribution. The probability of misclassification is much lower given their large number in the dataset. Conversely, anomalies are more challenging to detect as they are less frequent.



(a) On the Arrhythmia dataset.

(b) On the Thyroid dataset.

Figure 2: F1-score computed on different classes of interest.

[2] houssamzenati/Adversarially-Learned-Anomaly-Detection (github.com)

Report both precision and recall scores for better interpretability. Most research papers reviewed display both recall and precision scores, but too many still report only the F1-score. F1-score is the harmonic mean between precision and recall. Therefore, a good score can mask a poor recall with excellent precision or vice versa. For instance, DROCC's excellent 0.996 recall score on CSE-CIC-IDS2018 indicates that the model doesn't miss a lot of anomalies (low false-negative rate), but its 0.296 precision suggests it often flags normal samples as anomalies (high false-positive rate). This observation is not possible using only the F1-score (0.456).

AUPR is more informative than AUROC. The differences between AUROC and AUPR for all the models on the KDD and NSL-KDD are negligible. However, they differ significantly on CSE-CIC-IDS2018 where we see a significant drop (over 10%) in performance between the two metrics. DAE drops from 0.882 to 0.689, DSEBM-r from 0.727 to 0.398 and ALAD from 0.856 to 0.615 to name a few. This further demonstrates our case that AUROC gives an optimistic perspective on a classifier's performance by giving equal weights to predictions on abnormal and normal instances. In anomaly detection, we are interested in the performance on the anomalies and so it makes sense to use a metric more sensitive to predictions on that class.

Testing on KDD is insufficient. KDD and its 10 percent variant NSL-KDD should only be used as a kind of basic sanity check since they do not provide distinctive insights into the performance of the methods. All implemented models – both deep and shallow – perform exceptionally well on this dataset, with F1-scores above 0.90. We can therefore conclude that KDD is highly trivial for anomaly detection. Instead, datasets such as CSE-CIC-IDS2018 should be preferred for comparison in the area of network intrusion detection, as most models report poor performance and results vary considerably on this more challenging dataset. Also, CSE-CIC-IDS2018 simulates a more realistic network, unlike KDD which was built over 20 years ago.

Summary. Among the surprising results, we note that our vanilla auto-encoder DAE outperforms more sophisticated reconstruction-based methods like DAGMM and MemAE on CSE-CIC-IDS2018. The large number of samples and intra-class variation on the same dataset could explain the downfall of DeepSVDD, DROCC, and one-class classification approaches in general. More generally, baseline methods with optimized hyper-parameters achieve more competitive F1-scores than reported in the literature so far. On Arrhythmia and Thyroid, they even outperform most of their deep-learning counterparts. NeuTraLAD, the transformation-based approach, offers consistently above-average performance across all datasets. The data-augmentation strategy is particularly efficient on small-scale datasets where samples are scarce. The only adversarial approach (ALAD) does not distinguish itself significantly from the other reconstruction-based methods, which is to be expected since it uses a reconstruction objective as its core.

5 Conclusion

In this paper, we emphasize the importance of using a reliable evaluation protocol to assess anomaly detection methods, revealed inconsistencies in reported results and claims, proposed a consistent evaluation protocol, and provided an updated evaluation of the twelve popular unsupervised anomaly detection methods on five widely used tabular datasets. The results reported in this paper give a better understanding of the current standing of the various unsupervised anomaly detection methods for tabular data.

We addressed the sources of inconsistencies in the evaluation protocols and offered a solution for each. We solve the data split problem by training on normal data only and using all of the anomalous samples in the test set. We discussed how the choices of performance metrics must be mindful of the imbalance in typical anomaly detection datasets. We advocate for the F1-score, precision, recall, and AUPR metrics to be reported to give a better picture of the performances of each method. The strategy for choosing the threshold for classification must be the same in all the evaluated methods, which we fixed to using the optimal threshold. We also mentioned how the choice of the class of interest can skew the evaluations and we set it to be the minority class. Finally, implementation details are vital to reproduce results.

We consider these results as preliminary and hope to extend our study to non-tabular data, especially time series and image datasets. We also hope that future work can compare their empirical results with ours following the same evaluation protocol and ultimately improve the consistency and reliability of future comparisons.

REFERENCES

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C Courville, and Marc Bellemare. Deep reinforcement learning at the edge of the statistical precipice. *Advances in Neural Information Processing Systems*, 34, 2021.

Liron Bergman and Yedid Hoshen. Classification-based anomaly detection for general data. *arXiv preprint arXiv:2005.02359*, 2020.

Xavier Bouthillier, Pierre Delaunay, Mirko Bronzi, Assya Trofimov, Brennan Nichyporuk, Justin Szeto, Nazanin Mohammadi Sepahvand, Edward Raff, Kanika Madan, Vikram Voleti, et al. Accounting for variance in machine learning benchmarks. *Proceedings of Machine Learning and Systems*, 3:747–769, 2021.

Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. Lof: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pp. 93–104, 2000.

Thyago P Carvalho, Fabrízzio AAMN Soares, Roberto Vita, Roberto da P Francisco, João P Basto, and Symone GS Alcalá. A systematic literature review of machine learning methods applied to predictive maintenance. *Computers & Industrial Engineering*, 137:106024, 2019.

Raghavendra Chalapathy and Sanjay Chawla. Deep learning for anomaly detection: A survey. *arXiv preprint arXiv:1901.03407*, 2019.

Yang Chen, Nami Ashizawa, Chai Kiat Yeo, Naoto Yanai, and Seanglidet Yean. Multi-scale self-organizing map assisted deep autoencoding gaussian mixture model for unsupervised intrusion detection. *Knowledge-Based Systems*, 224:107086, 2021.

Zhaomin Chen, Chai Kiat Yeo, Bu Sung Lee, and Chiew Tong Lau. Autoencoder-based network anomaly detection. In *2018 Wireless Telecommunications Symposium (WTS)*, pp. 1–5. IEEE, 2018.

Davide Chicco and Giuseppe Jurman. The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC genomics*, 21(1):1–13, 2020.

CIC. Ids 2018 — datasets — research — canadian institute for cybersecurity — unb. `https://www.unb.ca/cic/datasets/ids-2018.html`, a. Online; accessed 13 March 2022.

CIC. Nsl-kdd — datasets — research — canadian institute for cybersecurity — unb. `https://www.unb.ca/cic/datasets/nsl.html`, b. Online; accessed 13 March 2022.

Damien Fourure, Muhammad Usama Javaid, Nicolas Posocco, and Simon Tihon. Anomaly detection: How to artificially increase your f1-score with a biased evaluation protocol. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 3–18. Springer, 2021.

Dong Gong, Lingqiao Liu, Vuong Le, Budhaditya Saha, Moussa Reda Mansour, Svetha Venkatesh, and Anton van den Hengel. Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1705–1714, 2019.

Kyle Gorman and Steven Bedrick. We need to talk about standard splits. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 2786–2791, 2019.

Sachin Goyal, Aditi Raghunathan, Moksh Jain, Harsha Vardhan Simhadri, and Prateek Jain. Drocc: Deep robust one-class classification. In *International Conference on Machine Learning*, pp. 3711–3721. PMLR, 2020.

László A Jeni, Jeffrey F Cohn, and Fernando De La Torre. Facing imbalanced data–recommendations for the use of performance metrics. In *2013 Humaine association conference on affective computing and intelligent interaction*, pp. 245–251. IEEE, 2013.

Rudolf Kadlec, Ondrej Bajgar, and Jan Kleindienst. Knowledge base completion: Baselines strike back. *arXiv preprint arXiv:1705.10744*, 2017.

Donghwoon Kwon, Hyunjoo Kim, Jinoh Kim, Sang C Suh, Ikkyun Kim, and Kuinam J Kim. A survey of deep learning-based network anomaly detection. *Cluster Computing*, 22(1):949–961, 2019.

Joffrey L Leevy, John Hancock, Richard Zuech, and Taghi M Khoshgoftaar. Detecting cybersecurity attacks across different network features and learners. *Journal of Big Data*, 8(1):1–29, 2021.

Tangqing Li, Zheng Wang, Siying Liu, and Wen-Yan Lin. Deep unsupervised anomaly detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 3636–3645, 2021.

Benjamin Marie, Atsushi Fujita, and Raphael Rubino. Scientific credibility of machine translation research: A meta-evaluation of 769 papers. *arXiv preprint arXiv:2106.15195*, 2021.

Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. A metric learning reality check. In *European Conference on Computer Vision*, pp. 681–699. Springer, 2020.

ODDS. Arrhythmia dataset – odds (stonybrook.edu). `http://odds.cs.stonybrook.edu/arrhythmia-dataset/`, a. Online; accessed 13 March 2022.

ODDS. Thyroid disease dataset – odds (stonybrook.edu). `http://odds.cs.stonybrook.edu/thyroid-disease-dataset/`, b. Online; accessed 13 March 2022.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Joelle Pineau, Philippe Vincent-Lamarre, Koustuv Sinha, Vincent Larivière, Alina Beygelzimer, Florence d'Alché Buc, Emily Fox, and Hugo Larochelle. Improving reproducibility in machine learning research: a report from the neurips 2019 reproducibility program. *Journal of Machine Learning Research*, 22, 2021.

Chen Qiu, Timo Pfrommer, Marius Kloft, Stephan Mandt, and Maja Rudolph. Neural transformation learning for deep anomaly detection beyond images. In *International Conference on Machine Learning*, pp. 8703–8714. PMLR, 2021.

Edward Raff. A step toward quantifying independently reproducible machine learning research. *Advances in Neural Information Processing Systems*, 32, 2019.

Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In *International conference on machine learning*, pp. 4393–4402. PMLR, 2018.

Lukas Ruff, Jacob R Kauffmann, Robert A Vandermeulen, Grégoire Montavon, Wojciech Samek, Marius Kloft, Thomas G Dietterich, and Klaus-Robert Müller. A unifying review of deep and shallow anomaly detection. *Proceedings of the IEEE*, 2021.

Takaya Saito and Marc Rehmsmeier. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PloS one*, 10(3):e0118432, 2015.

Bernhard Schölkopf, Robert C Williamson, Alex Smola, John Shawe-Taylor, and John Platt. Support vector method for novelty detection. *Advances in neural information processing systems*, 12, 1999.

Erich Schubert, Arthur Zimek, and Hans-Peter Kriegel. Local outlier detection reconsidered: a generalized view on locality with applications to spatial, video, and network outlier detection. *Data mining and knowledge discovery*, 28(1):190–237, 2014.

Swee Chuan Tan, Kai Ming Ting, and Tony Fei Liu. Fast anomaly detection for streaming data. In *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.

Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali A Ghorbani. A detailed analysis of the kdd cup 99 data set. In *2009 IEEE symposium on computational intelligence for security and defense applications*, pp. 1–6. Ieee, 2009.

Alaa Tharwat. Classification assessment methods. *Applied Computing and Informatics*, 2020.

Qi Wei, Yinhao Ren, Rui Hou, Bibo Shi, Joseph Y Lo, and Lawrence Carin. Anomaly detection for medical images based on a one-class classification. In *Medical Imaging 2018: Computer-Aided Diagnosis*, volume 10575, pp. 375–380. SPIE, 2018.

Yi-Xuan Xu, Ming Pang, Ji Feng, Kai Ming Ting, Yuan Jiang, and Zhi-Hua Zhou. Reconstruction-based anomaly detection with completely random forest. In *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*, pp. 127–135. SIAM, 2021.

Houssam Zenati, Manon Romain, Chuan-Sheng Foo, Bruno Lecouat, and Vijay Chandrasekhar. Adversarially learned anomaly detection. In *2018 IEEE International conference on data mining (ICDM)*, pp. 727–736. IEEE, 2018.

Shuangfei Zhai, Yu Cheng, Weining Lu, and Zhongfei Zhang. Deep structured energy based models for anomaly detection. In *International conference on machine learning*, pp. 1100–1109. PMLR, 2016.

Chong Zhou and Randy C Paffenroth. Anomaly detection with robust deep autoencoders. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 665–674, 2017.

Xiaoqian Zhu, Xiang Ao, Zidi Qin, Yanpeng Chang, Yang Liu, Qing He, and Jianping Li. Intelligent financial fraud detection practices in post-pandemic era. *The Innovation*, 2(4):100176, 2021.

Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *International conference on learning representations*, 2018.

APPENDIX

A MODELS DESCRIPTION

We chose a set of methods from highly cited papers published between 2016 and the time of this publishing and models considered as baseline. We chose baseline models based on the reputation of their paper, number of citations and number of times they are considered as baseline. We also include more recent models that claim state-of-the-art performance on a subset of the datasets previously described.

Deep Structured Energy Based Models for Anomaly Detection Zhai et al. (2016). DSEBM performs density estimation through energy-based models. The energy function is composed of neural networks and energy is accumulated across the multiple layers Zenati et al. (2018). Two different anomaly scores are studied: reconstruction error (DSEBM-r) and energy score (DSEBM-e).

Deep Auto Encoding Gaussian Mixture Model Zong et al. (2018). DAGMM trains an autoencoder and a feed-forward network in an end-to-end fashion. The reconstruction error and the latent representation produced by the autoencoder are given as input to a MLP which is used to estimate the parameters of a Gaussian Mixture Model (GMM). The output of the later network is ultimately used to compute the log-likelihood of the samples, which is then used as an anomaly score. While no official code is available for this model, the original paper provided enough information for us to reimplement it.

Memory-augmented Deep Autoencoder Gong et al. (2019). MemAE leverages the representational potential of an encoder with a memory module with a sparse attention-based addressing mechanism to record the prototypical patterns in the data. A decoder is used to reconstruct the original sample from the items retrieved from the memory module. High reconstruction errors are associated with anomalies. We reused the implementation offered by the authors on Github.

Deep One-Class Classification Ruff et al. (2018). DeepSVDD leverages the representational potential of deep neural networks to learn a representation of the data that encompasses a hypersphere. By minimizing the volume of this hypersphere,the network is encourage to extract the common factors of variation in the training data. Points outside the hypersphere are predicted as anomalies. Fortunately, a PyTorch implementation is made available by the authors.

Deep Robust One-Class Classification Goyal et al. (2020). DROCC assumes that the normal points lie on a low-dimensional manifold that is well sampled. Based on this assumption, the authors develop a method that can train deep neural network architecture by generating anomalous points. For each normal point, gradient ascent is used to generate anomalous points that maximize the loss of the network. Using this adversarial approach, DROCC is able to synthetically generate data to train the DNN architecture in a supervised manner. We integrated the authors' PyTorch code (https://github.com/microsoft/EdgeML/).

Aversarially Learned Anomaly Detection Zenati et al. (2018). ALAD expands on the GAN foundations by adding an encoder to map data points to the latent space, and two more discriminators to ensure data-space and latent-space cycle-consistencies. ALAD uses the reconstruction error based on features extracted from an intermediate layer of a discriminator as the anomaly score.

Neural Transformation Learning for Deep Anomaly Detection Beyond Images Qiu et al. (2021). NeuTraLAD leverages the recent success of contrastive learning in images and adapts it to tabular data. It uses a data augmentation scheme with a deterministic contrastive loss. This scheme encourages transformed samples to be similar to the original input while encouraging dissimilarity between the transformed samples. Instead of using predefined transformations (such as rotation, translation, cropping, etc.) that are well-suited for computer vision tasks, NeuTraLAD learns the data transformations with a set of neural networks.

The previous SOTA methods are complemented with two of the most recurring baseline methods found within the anomaly detection literature.

One-Class SVM Schölkopf et al. (1999). OC-SVM is a popular one-class classification SVM algorithm used for anomaly detection. We used the Scikit-Learn implementation and experimented on different ν. The other parameters were set as defaults.

Local Outlier Factor Breunig et al. (2000). LOF classifies samples with substantially lower density than their neighbors as anomalies. The Scikit-Learn version was again used with optimized values for `n_neighbors`.

B HYPERPARAMETERS

In this appendix we describe the hyperparameters used to obtain the reported results.

	ALAD				
	Batch	Epoch	Lat. dim.	Weight decay	Learning rate
Arrhythmia	128	10000	32	0.0001	0.0001
Thyroid	128	20000	32	0.0001	0.0001
KDDCUP 10	1024	100	32	0.0001	0.0001
NSL-KDD	1024	200	32	0.0001	0.0001
CSE-CIC-IDS2018	1024	150	32	0.0001	0.0001

Table 4: ALAD hyperparameters.

	DAE			
	Batch	Epoch	Lat. dim.	Learning rate
Arrhythmia	128	10000	3	0.0001
Thyroid	128	5000	2	0.0001
KDDCUP 10	1024	100	2	0.0001
NSL-KDD	1024	100	2	0.0001
CSE-CIC-IDS2018	1024	100	2	0.0001

Table 5: DAE hyperparameters.

	DAGMM				
	Batch	Epoch	Lat. dim.	Learning rate	Weight decay
Arrhythmia	128	10000	2	0.0001	0.0001
Thyroid	128	5000	2	0.0001	0.0001
KDDCUP 10	1024	200	1	0.0001	0.0001
NSL-KDD	1024	200	1	0.0001	0.0001
CSE-CIC-IDS2018	1024	100	1	0.0001	0.0001

Table 6: DAGMM hyperparameters.

	DSEBM				
	Batch	Epoch	Lat. dim.	Learning rate	Weight decay
Arrhythmia	128	10000	2	0.0001	0.0001
Thyroid	128	5000	2	0.0001	0.0001
KDDCUP 10	1024	100	512	0.0001	0.0001
NSL-KDD	1024	100	512	0.0001	0.0001
CSE-CIC-IDS2018	1024	100	512	0.0001	0.0001

Table 7: DSEBM hyperparameters.

	DeepSVDD	
	Batch size	Number of output features
Arrhythmia	128	64
Thyroid	128	1
KDDCUP 10	1024	29
NSL-KDD	1024	31
CSE-CIC-IDS2018	1024	16

Table 8: DeepSVDD hyperparameters.

	DROCC							
	Batch size	Threshold	Radius	μ	ν	Learning rate	Only CE epochs	Gradient ascent steps
Arrhythmia	256	70	16	0.5	0.1	0.01	50	50
Thyroid	256	95	0.5	1.0	0.01	0.0001	50	50
KDDCUP 10	1024	61	16	0.5	0.1	0.01	50	50
NSL-KDD	-1	35	16	0.5	0.1	0.01	50	50
CSE-CIC-IDS2018	100	2	8.124	1.0	0.01	0.0001	50	50

Table 9: DROCC hyperparameters.

	DUAD							
	Batch	Epoch	Lat. dim.	r	p_0	p_s	Clusters	Learning rate
Arrhythmia	128	5000	3	10	35	30	8	0.0001
Thyroid	128	5000	2	10	35	30	8	0.0001
KDDCUP 10	1024	100	2	10	35	30	10	0.0001
NSL-KDD	1024	100	2	10	35	30	10	0.0001
CSE-CIC-IDS2018	1024	100	2	10	35	30	15	0.0001

Table 10: DUAD hyperparameters.

	MemAE					
	Batch	Epoch	Lat. dim.	Mem. dim.	Weight decay	Learning rate
Arrhythmia	128	10000	3	50	0.0001	0.0001
Thyroid	128	20000	3	50	0.0001	0.0001
KDDCUP 10	1024	200	3	50	0.0001	0.0001
NSL-KDD	1024	200	3	50	0.0001	0.0001
CSE-CIC-IDS2018	1024	50	3	250	0.0001	0.0001

Table 11: MemAE hyperparameters.

	NeuTraLAD					
	Batch	Epoch	Lat. dim.	Learning rate	Weight decay	Transformation type
Arrhythmia	128	200	32	0.0001	0.00001	residual
Thyroid	128	580	24	0.0001	0.00001	residual
KDDCUP 10	1024	40	32	0.0001	0.00001	multiplicative
NSL-KDD	1024	40	32	0.0001	0.00001	multiplicative
CSE-CIC-IDS2018	1024	25	32	0.0001	0.00001	multiplicative

Table 12: NeuTraLAD hyperparameters.

	SOM-DAGMM			
	Batch	Epoch	Lat. dim.	Learning rate
Arrhythmia	128	10000	2	0.0001
Thyroid	128	5000	1	0.0001
KDDCUP 10	1024	100	2	0.0001
NSL-KDD	1024	100	2	0.0001
CSE-CIC-IDS2018	1024	100	2	0.0001

Table 13: SOM-DAGMM hyperparameters.

	OC-SVM		LOF	
	Threshold	ν	Threshold	Number of neighbors
Arrhythmia	73	0.40	75	50
Thyroid	97	0.05	96	20
KDDCUP 10	78	0.25	77	100
NSL-KDD	46	0.40	44	20
CSE-CIC-IDS2018	86	0.01	88	15

Table 14: OC-SVM and LOF hyperparameters.