

# Deep Learning for Network Anomaly Detection under Data Contamination: Evaluating Robustness and Mitigating Performance Degradation

D’Jeff K. Nkashama<sup>[0000–0002–5541–7837]</sup>, Jordan F.  
Masakuna<sup>[0000–0001–6165–8402]</sup>, Arian Soltani<sup>[0009–0001–6967–5730]</sup>, Jean-Charles  
Verdier, Marc Frappier<sup>[0000–0002–4402–2514]</sup>, Pierre-Martin  
Tardif<sup>[0000–0002–7413–6897]</sup>, and Froduald Kabanza<sup>[0009–0007–6205–6447]</sup>

Université de Sherbrooke, QC J1K 2R1, Canada  
`nkad2101@usherbrooke.ca`

**Abstract.** Deep learning (DL) has emerged as a crucial tool in network anomaly detection (NAD) for cybersecurity. While DL models for anomaly detection excel at extracting features and learning patterns from data, they are vulnerable to data contamination—the inadvertent inclusion of attack-related data in training sets presumed benign. This study evaluates the robustness of six unsupervised DL algorithms against data contamination using our proposed evaluation protocol. Results demonstrate significant performance degradation in state-of-the-art anomaly detection algorithms when exposed to contaminated data, highlighting the critical need for self-protection mechanisms in DL-based NAD models. To mitigate this vulnerability, we propose an enhanced auto-encoder with a constrained latent representation, allowing normal data to cluster more densely around a learnable center in the latent space. Our evaluation reveals that this approach exhibits improved resistance to data contamination compared to existing methods, offering a promising direction for more robust NAD systems.

**Keywords:** Unsupervised learning · deep learning · robustness · anomaly detection · intrusion detection systems · IoT network

## 1 Introduction

In today’s digital age, information technology has become an integral part of our daily activities, including communication, travel, work, banking, education, and manufacturing. An illustrative example of this integration is the Internet of Things (IoT), marking a substantial change in our interaction with technology [41]. IoT allows remote monitoring and control, bridging the gap between the physical world and computer-based systems. While technology offers numerous benefits, it can also be exploited by adversaries to conduct malicious activities on cyberinfrastructure [16,6]. These malicious activities can take various forms,

including information theft, denial of service, and data contamination, among others, posing threats to operational cyber systems. Consequently, the need to safeguard cyberinfrastructures against these persistent and destructive cyber threats is of paramount importance.

Recent advances in machine learning (ML) or deep learning (DL) have played a fundamental role and made remarkable progress in the field of cybersecurity at a rapid pace [24,29,10,12]. Numerous ML-based anomaly detection (AD) techniques have found practical use in the realm of cybersecurity [24,43,10,12]. An anomaly represents an observation significantly deviating from what is considered normal behavior [1]. Based on the context, such observations can be deemed unusual, irregular, inconsistent, unexpected, faulty, fraudulent, or malicious [38,9]. Examples of AD applications in cybersecurity include malware detection, spam filtering, and intrusion detection systems (IDS) [12,29,33,47,14].

Training ML models for AD requires a large amount of data usually collected from various sources (e.g. sensors, actuators, and processes). However, as the volume of data grows, it becomes increasingly vulnerable to contamination [22,6,7,31,35]. There are several potential sources of data contamination. One source of contamination could be an ongoing attack campaign, which may go unnoticed during data collection, leading to the inclusion of malicious instances in the training data. Misconfigurations in equipment are another source of contamination. Moreover, adversaries, aware of the importance of data in ML-based IDS, could attempt to deliberately inject malicious samples into the training set. The adversaries' goal is to hamper model performance and possibly create a backdoor for subsequent intrusions. For example, in a network traffic analysis within an IoT environment, data contamination can cause a DL model to erroneously predict traffic related to a Denial of Service (DoS) attack as legitimate user activity [33,47]. This misclassification has the potential to disrupt the availability of services. Furthermore, an attacker may compromise a device by gaining remote control and injecting carefully crafted traffic from an external network. While evasion attacks during testing [4,33] are also a concern, our research emphasizes the importance of mitigating data contamination during the model's training process to enhance robustness against data contamination.

In the current landscape of AD model training, it is typically assumed that the training data is clean and free of contamination. However, ensuring the cleanliness of data can be a resource-intensive process, especially when dealing with large-scale datasets [21]. Many AD models are not routinely subjected to data verification procedures. This becomes particularly critical in intrusion detection, where the resilience of AD models against data contamination is crucial. Although the robustness of AD models has already been studied, the most recent AD models are rarely exposed to and tested on contaminated data [36,6]. This lack of exposure to real-world data contamination can be a significant limitation when considering the practicability of these models. Effectively addressing this challenge involves developing robust AD models.

This paper evaluates the impact of training data contamination on the performance of six unsupervised deep learning models for network AD. We chose

these models because they are recent deep learning-based AD methods, selected for their superior performance compared to their predecessors [36,10,50,48], and the diversity in their approaches to modeling the AD task. These approaches include self-supervised [36], generative adversarial [48], energy-based [49], density modeling [50], and reconstruction-based [26,50,10] methods. It is important to note that the data contamination discussed in this study involves injecting malicious instances into the training set, which is expected to be cleaned for the AD task, while keeping their features unchanged.

#### **Contributions:**

In this paper, our contribution is threefold. Firstly, we propose an evaluation protocol that integrates data contamination to emulate a real-world challenge in training DL for unsupervised AD task. The protocol enables fair and rigorous assessment of models’ robustness and prevents data leakage during the evaluation process. We employ this protocol to evaluate the robustness of six state-of-the-art DL models for unsupervised AD: ALAD [48], Deep auto-encoder [10], DAGMM [50], DSEBM [49], DUAD [26], and NeuTraLAD [36]. Through a series of experiments, we systematically analyze the performance of the selected models under various levels of data contamination. We use the following cybersecurity benchmark datasets: CIC-CSE-IDS2018, Kitsune, CIC-IoT23, KDDCUP, and NSL-KDD<sup>1</sup>.

Secondly, we enhance the performance of a standard deep autoencoder (DAE) by adding a constraint that data in its latent space should be more densely concentrate around a learnable center  $c$ . We refer to the improved models as DAE-LR and DUAD-LR. Our experiments demonstrate significant improvements in overall performance, particularly when dealing with data contamination, compared to standard versions of autoencoders, a mostly used model in network AD [7,25,6].

Lastly, through an extensive experiments, we underscore the potential pitfalls of relying on outdated cybersecurity datasets for evaluating model performance. We demonstrate how model performance on such datasets can be misleading when assessing their effectiveness in current network environments with evolving attack patterns. We also emphasize the significance of robustness to data contamination as a critical criterion when selecting an AD model for cybersecurity applications. Our study provides valuable guidance to practitioners and researchers in choosing models that can withstand data contamination commonly encountered in real-world intrusion detection scenarios.

## **2 Related Work and Background**

### **2.1 Deep Unsupervised Anomaly Detection**

Unsupervised anomaly detection consists of separating normal data from anomalies or learning the underlying data distribution of normal data with no access

---

<sup>1</sup> <https://www.unb.ca/cic/datasets/nsl.html>

to labels. This approach is challenging since the absence of explicit labels limits exploitable information.

Unsupervised Anomaly detection (UAD) approaches can be broadly categorized into four types. (1) Probabilistic methods, such as DAGMM [50], estimate the probability density function of the data and flag samples lying in the low-density region as anomalies. (2) Reconstruction-based methods aim to approximate an identity function of input data, assuming that normal data is compressible and can be reconstructed accurately. Based on the reconstruction error—the scoring function—the learned model flags data samples that cannot be reconstructed well from their compression as anomalous. Examples of these methods include MemAE [18] and DUAD [26]. (3) Distance-based models, such as LOF [8] and iForest [27], predict anomalies by measuring their distances from normal data, with unusual samples being those with significantly different distances. (4) One-class classification approaches, such as OC-SVM [42] and DeepSVDD [39], try to identify normal data points amongst all data, primarily by learning the underlying data distribution of only normal data. Each of these categories has its own strengths and weaknesses, making them suitable for different types of AD tasks.

Classical approaches to unsupervised AD work well with lower dimensions. However, as the feature space grows, they suffer from the curse of dimensionality [26]. Deep learning architectures, for instance, autoencoders [10] have shown significant potential to mitigate this issue [44,50].

In this paper, we evaluate the robustness of various unsupervised deep learning methods against data contamination. We selected these methods based on their recent developments, superior performance compared to their predecessors, and the diversity of their approaches to modeling the AD task, including reconstruction-based, energy-based, generative-based, and self-supervised-based approaches [48,10,50,49,26,36]. No such study, with a wide range of approaches, on network AD has been conducted as far as we know, yet resilience to data contamination should be an important requirement of deep learning-based network AD methods.

## 2.2 Data contamination

Training a network AD model requires data containing only benign traffic. However, in reality, collected data may contain both benign and malicious instances, resulting in a contaminated dataset. Contaminated instances may lead to higher prediction errors during testing and deployment [2]. There are several causes of these corrupted instances, including misconfiguration, ongoing attacks during data collection, or adversaries attempting to manipulate the training process to create backdoors.

Research on data contamination has gradually surged, leading to the development of various techniques. Earlier works focused on SVMs [3], feature selection [47], and PCA [37], as well as initial deep learning efforts that mainly addressed reducing false classification [32]. Subsequently, research in this area shifted toward generative models [19], where attacks could create a backdoor, allowing

samples with specific and easily manipulated characteristics to evade detection by the model [46]. Most of these studies primarily focused on data contamination in image classification [3,4,15], linear regression [23], and IoT environments [6]. However, little attention has been paid to data contamination in the context of network intrusion detection systems, especially considering the recent advancements and high-performing unsupervised deep learning models for AD, as well as the mitigation of model performance degradation. In this study, we evaluate the robustness of models against data contamination and propose a latent regulated approach to improve the performance of deep autoencoders.

### 3 Method

#### 3.1 Latent Regulated Deep Auto-Encoder

In Figure 1, we present the architecture of the Deep Autoencoder (DAE), which consists of two components: an encoder and a decoder. The encoder’s role is to transform the input data into a low-dimensional representation, referred to as the latent representation. In contrast, the decoder’s function is to reconstruct the original input data from this latent representation. Training this model is done by minimizing the reconstruction error in Equation (1). Furthermore, the same reconstruction error also serves as the anomaly scoring function.

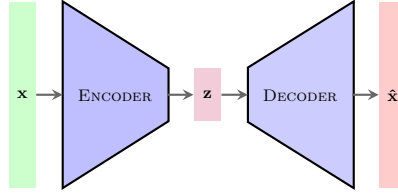


Fig. 1: Deep Auto-Encoder architecture where,  $x$ ,  $z$ , and  $x'$  denote the input data, the latent representation, and the reconstructed input, respectively.

$$L_{\theta}(x) = \|x - \hat{x}\|^2. \quad (1)$$

where,  $\|\cdot\|$  is the  $\ell_2$ -norm.

The standard version of DAE allows flexibility in how the model represents data in the latent space, focusing primarily on minimizing the reconstruction error. However, when dealing with contaminated samples, controlling this latent space projection is crucial. Ideally, normal data should be clustered densely in the latent space, making the representation of anomalies more dispersed. To achieve this, we introduce an additional constraint ( $\|z - c\|^2 < \epsilon$ ) to regularize the latent representation.

The optimization formulation for the DAE-LR is then given by:

$$\min_{\theta} \|x - \hat{x}\|^2 \quad \text{subject to} \quad \|z - c\|^2 \leq \epsilon, \quad (2)$$

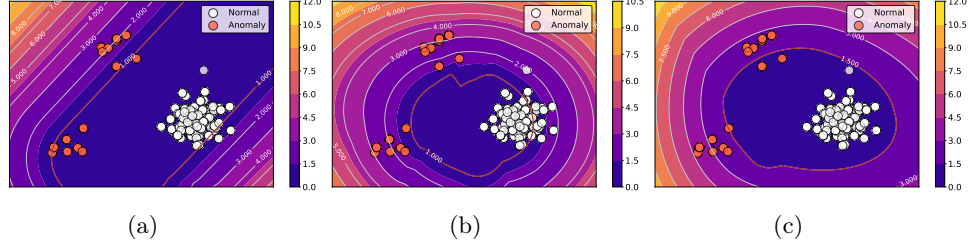


Fig. 2: Contour plot showing the decision boundary of DAE-LR utilizing our proposed loss function from Equation (3), and trained on 2D synthetic contaminated data with different types of center  $c$ . (a) Standard DAE ( $\lambda = 0$ ). (b) DAE-LR mean-centered ( $c = \bar{z}$ ). (c) DAE-LR with a learnable center.

where  $x$  is the input data,  $\hat{x}$  is the reconstructed data,  $z$  is the latent representation,  $c$  is a learnable center in the latent space, and  $\epsilon$  is a small threshold that controls the clustering of normal data around  $c$ .

The problem formulated in Equation (2) leads to the following loss function, which should be minimized with respect to the parameters  $\theta$  and  $c$ :

$$L_{\theta}^{(\text{ours})}(x) = \|x - \hat{x}\|^2 + \lambda \|z - c\|^2. \quad (3)$$

where,  $z = \text{encoder}(x)$  is the latent representation of the input data,  $\theta$  and  $c$  are models' parameters, and  $\lambda > 0$  the regularizing factor for the latent representation.

Notably, the combination of latent constraint term and reconstruction error helps control how the input data is represented in the latent space, enforcing it to be centered around a center  $c$ . Figure 2c shows how this combination contributes to a better decision boundary compared to the standard version in Figure 2a, when the training data is contaminated.

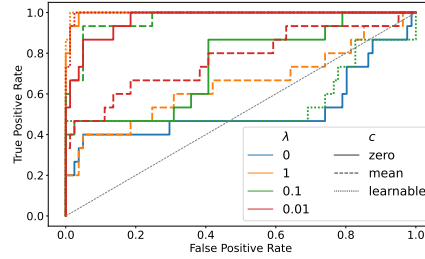


Fig. 3: ROC curves illustrating the performance of the latent-regulated DAE-LR (as described in Equation 3), which has been trained on 2D synthetic and contaminated data. It's important to note that when  $\lambda = 0$ , it corresponds to the standard DAE version.

There are various choices for the center,  $c$ . A trivial choice could be to set  $c = \mathbf{0}$ , or to use the average of latent representations (i.e.,  $c = \bar{z} = \frac{1}{n} \sum_{i=1}^n z_i$ ). Instead of making such strong assumptions on what the center  $c$  should be, we allow the model to determine the optimal center during training. The center  $c$  is a learnable parameter.

Figure 2b and 2c display the decision boundary using a mean-centered and learnable-centered latent representation, respectively. We observe that using mean-centered may lead to increasing false positive alerts, while the learnable-centered  $c$  produces a better decision boundary, confirming our intuition. Furthermore, Figure 3 displays ROC curves with different choices of  $\lambda$  and the center  $c$ . We can see that a learnable  $c$  yields better performance.

## 4 Proposed Evaluation Protocol

This section presents the protocol outlined in Algorithm 1, which is used to evaluate the robustness of the selected methods. We aim to address the following question concerning our proposed evaluation protocol: *How can we effectively ensure a fair and rigorous model evaluation under contamination, capture variance across runs, mitigate the risk of data leakage, and accommodate the class imbalances typically observed in network intrusion detection datasets?* Furthermore, we investigate the added value of incorporating critical difference diagrams to enhance model performance ranking and facilitate comprehensive pairwise comparisons in this specific context. Figure 4 illustrates the workflow for one run using this evaluation protocol.

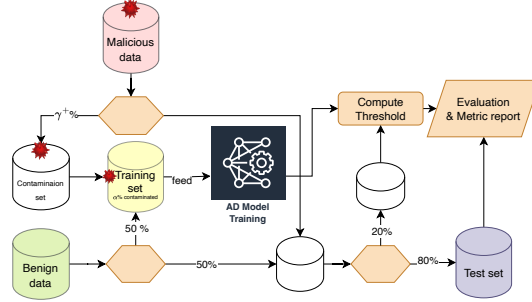


Fig. 4: Proposed evaluation protocol workflow for a single run. The model is trained with a contaminated set characterized by contamination ratio  $\alpha$ . The proportion of benign traffic and attack data in the test set is maintained constant across runs, regardless of the level of training set contamination. Moreover, the model threshold for the anomaly scoring function is computed on a separate validation set, distinct from the final test set. The final test set is excluded from the training process to prevent data leakage.

**Algorithm 1** Evaluation Protocol

**Require:**  $X$ ,  $\gamma_-$ ,  $\gamma_+$ ,  $g_\theta$ ,  $\alpha_0$ , and  $\alpha_1$  represent the dataset, the proportion of normal data for the training set, the proportion of anomalous data for the contamination set, an increment of contamination rate, and the maximum value of the contamination rate, respectively.

**function** EVALUATION-PROTOCOL( $X, \gamma_+, \gamma_-, g_\theta, \alpha_0, \alpha_1$ )

```

 $X^+ \leftarrow \text{attack-traffic}(X)$ 
 $X^- \leftarrow \text{benign-traffic}(X)$ 
metrics  $\leftarrow \{\text{recall, precision, f1-score}\}$ 
 $N^+ \leftarrow \#X^+$ 
 $N^- \leftarrow \#X^-$ 
 $M^+ \leftarrow \lceil (1 - \gamma_+)N^+ \rceil$ 
 $M^- \leftarrow \lceil (1 - \gamma_-)N^- \rceil$ 
 $X_{\text{train}}^+ \leftarrow \text{random-selection}(X^+, M^+)$ 
 $X_{\text{train}}^- \leftarrow \text{random-selection}(X^-, M^-)$ 
 $X_{\text{test}}^+ \leftarrow X^+ \setminus X_{\text{train}}^+$ 
 $X_{\text{test}}^- \leftarrow X^- \setminus X_{\text{train}}^-$ 
 $X_{\text{test}} \leftarrow X_{\text{test}}^+ \cup X_{\text{test}}^-$ 
 $X_{\text{threshd}} \leftarrow \text{random-selection}(X_{\text{test}}, \lceil .2 * \#X_{\text{test}} \rceil)$ 
 $X_{\text{test}} \leftarrow X_{\text{test}} \setminus X_{\text{threshd}}$ 
 $D_{\text{train}}^+, D_{\text{train}}^- \leftarrow \text{preprocessing}(X_{\text{train}}^+, X_{\text{train}}^-)$ 
 $D_{\text{test}}, D_{\text{threshd}} \leftarrow \text{preprocessing}(X_{\text{test}}, X_{\text{threshd}})$ 
 $n \leftarrow \#D_{\text{train}}^-$ 
 $\alpha \leftarrow 0$ 
scores  $\leftarrow []$ 
while  $\alpha \leq \alpha_1$  do
   $m \leftarrow \lceil \frac{\alpha n}{(1-\alpha)} \rceil$ 
   $C \leftarrow \text{random-selection}(D_{\text{train}}^+, m)$ 
   $D_{\text{train}} \leftarrow D_{\text{train}}^- \cup C$ 
   $g_\theta \leftarrow \text{train}(g_\theta, D_{\text{train}})$ 
  threshold  $\leftarrow \text{estimate\_threshold}(g_\theta, D_{\text{threshd}})$ 
  score  $\leftarrow \text{evaluation}(g_\theta, D_{\text{test}}, \text{metric, threshold})$ 
  scores.push(score,  $\alpha$ )
   $\alpha \leftarrow \alpha + \alpha_0$ 
end while
return scores
end function

```

**Data split strategy.** We split the dataset following the setup proposed by [50], the training set contains a proportion  $\gamma_-$  of normal data samples randomly drawn, but we add a ratio  $\alpha$  of *attack related data*. The test set contains the rest of the normal data plus a percentage  $(1 - \gamma_+)$  of attack data with  $\gamma_+ \in \{20, 40\}$ . The remaining  $\gamma_+$  percent of attack data is in a set that we call the *contamination* set. More specifically, the ratio  $\alpha$  of attack data utilized to contaminate the training set stems from that contamination set.

The underlying motivation for employing this specific data split strategy is to maintain both consistency between various sets of experiments and fairness



in the evaluation of diverse algorithms, as highlighted in [1]. By maintaining a constant proportion of normal and attack data within the test set, regardless of the training set contamination level, we facilitate a robust assessment of how varying levels of contamination in the training set impact the performance of models.

**Captured Variance.** It’s important to note that we consistently apply this data split strategy in every run, with different seeds for randomness, allowing us to capture the variance arising from data sampling, in addition to the variance attributed to model’s parameter initialization, as discussed in [5].

**Performance metrics and threshold.** We report the average and standard deviation of *F1-score*, *precision*, and *recall* over 20 runs for every ratio  $\alpha$  of the training set contamination. These metrics are suitable for problems with imbalanced class datasets [11]. Since network intrusion detection datasets have a significant class imbalance, we consider attack related data as the class of interest or the positive class. Attack data is often the minority class, while normal data is the majority class in a real-world application. The scarcity of attack data makes it crucial to evaluate the performance of models on this data. Furthermore, assessing model performance with normal data as the class of interest can overestimate the classifier and be misleading [40,1]. Therefore, all metrics are computed using the attack data class as the positive class. However, F1-score, precision, and recall calculation require setting a decision threshold on the samples’ anomaly scores. For that we propose to use 20% of the test set to find the decision threshold and then test it on the remaining 80%. We choose the decision threshold such that the F1-score – the harmonic mean of precision and recall – is the best that the model could achieve.

**Rigorous and leakage free model evaluation.** It is intuitive that creating disjoint subsets of data as described in the data split strategy mentioned above prevents some sort of data leakage during the determination of the model’s threshold or the final evaluation. Subsets created are training, contamination, and test sets. This strategy is data-leakage-free because there is no interaction with the final test set (which represents 80% of the test set) during the training phase. Such segmentation of data makes the model to be built on the training set with contamination data stemming from the contamination set, and the threshold defined using the 20% of the test set; the final test set remains entirely untouched and unknown to the model before the final evaluation. This ensures that the model’s performance is measured accurately on previously unseen data. It also eliminates any potential bias that may arise if the test set were used to adjust the model’s parameters or to determine the model’s threshold. Hence, the proposed evaluation protocol indicates that the model’s performance and robustness are evaluated objectively.

**Models comparison and ranking.** We additionally suggest the use of the critical difference diagram (CD diagram) [13] to derive the overall model performance ranking and to perform pairwise comparisons of models from a statistical standpoint. The CD diagram is constructed using the Wilcoxon-Holm method. The Wilcoxon-Holm method, which is a pairwise rank test, is recommended when

distributions of data being compared are not necessarily normal, and models are run from the same initial conditions which makes pairwise tests more appropriate. In addition, as mentioned above, the Wilcoxon-Holm method uses  $p$ -values to measure significant differences in data. However, it is mentioned that the  $p$ -value alone is not enough to describe the statistical difference between two groups of data [17] as one might not be able to quantify, from  $p$ -values only, how large statistical differences are. Another important metric to measure the significance of differences from two groups of samples needs to be considered—effect size. In other words, the  $p$ -value and the effect size should be combined for a more complete statistical conclusion.

## 5 Experimental investigation

We first describe the datasets and the models, and then discuss experimental results. The distributions of these datasets are visualized in Figure 5, and their characteristics are summarized in Table 1.

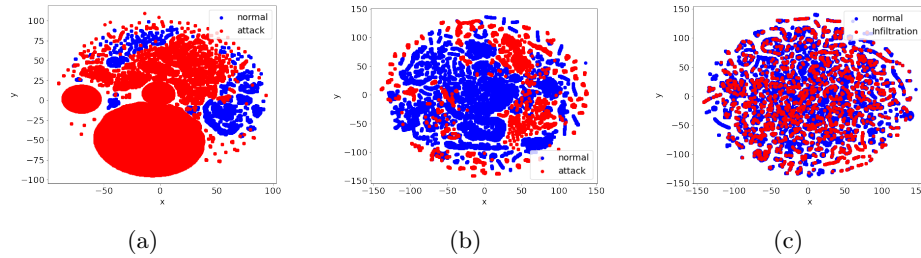


Fig. 5: Visualization of Normal Traffic and Attack Data in a Two-Dimensional Space Using t-SNE [30]. (a) and (b) Show normal and attack data from the KD-CUP and NSL-KDD datasets, respectively, with all attack types combined into a single class. (c) Displays normal data and specifically highlights the *infiltration* attack data from the CSE-CIC-IDS2018 dataset.

### 5.1 Datasets

Table 1 shows the statistics of datasets. We applied Min-Max scaling to the continuous features and one-hot encoding to the categorical features for all the datasets.

**KDDCUP**<sup>2</sup> is a cyber-intrusion detection dataset that is 20 years old. While it conveys well-known issues such as duplicated samples [45], it is still widely used as a benchmark dataset. It contains data related to normal traffic and attacks simulated in a military network environment. The Attacks include DoS, R2L,

<sup>2</sup> <http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html>

U2R, and probing. The dataset has 41 features, with 34 continuous, and the rest are categorical features. **NSL-KDD** is a revised version of the KDDCUP dataset provided by the Canadian Institute of Cybersecurity (CIC). **CIC-CSE-IDS2018** is also provided by CIC in collaboration with the Communication Security Establishment. Unlike the KDDCUP dataset, this dataset is recent, and it contains normal traffic plus attacks data simulated on a complex network. Attack types include Brute-force, Heartbleed, Botnet, DoS, DDoS, Web attacks, and infiltration of the network from the inside [43]. In our experiments, we utilized both the original and the revised version of this dataset, referred to as CIC-IDS2018R [28] in the experiment section.

**Kitsune**<sup>3</sup> is a cybersecurity dataset collected from a commercial IP-based surveillance system. This dataset is related to network traffic within an IoT (Internet of Things) environment. The types of attacks in this dataset include OS scans, fuzzing, video injection, SSDP flooding, and Botnet malware.

**CIC-IoT23** is a novel network traffic dataset designed specifically for IoT environments, serving as a benchmark for research and evaluation. The IoT network comprises a total of 105 devices, as detailed in [34]. Here the types of attacks it contains: DDoS and DoS, Recon, Brute Force, Spoofing, and Mirai.

Table 1: Datasets statistics.

Dataset	# Samples	# features	Attacks ratio
KDDCUP 10%	494 021	42	0.1969
NSL-KDD	148 517	42	0.4811
CSE-CIC-IDS2018	16 232 944	83	0.1693
Kitsune	25 758 272	115	0.2304
CIC-IoT23	4 671 520	41	0.2704

## 5.2 Baseline Models

We implemented and investigated six unsupervised AD models. Our source code can be found at [https://anonymous.4open.science/r/network\\_anomaly\\_detection\\_robustness-4EF3/README.md](https://anonymous.4open.science/r/network_anomaly_detection_robustness-4EF3/README.md).

**Deep Unsupervised Anomaly Detection [26].** DUAD is a method that employs a Deep Autoencoder (DAE) for AD. It is based on the hypothesis that anomalies in the training set are samples with relatively high variance distribution. Unlike a standard DAE, DUAD applies distribution clustering after a fixed number of iterations to select a subset of normal data from the training set—specifically, clusters with low variance. The reconstruction error is used as the anomaly score. We integrated the DUAD approach with our DAE-LR model, which we refer to as DUAD-LR.

<sup>3</sup> <https://archive.ics.uci.edu/dataset/516/kitsune+network+attack+dataset>

**Deep Structured Energy Based Models for Anomaly Detection [49].** DSEBM, as its name suggests, is an energy-based model. It learns the energy function of input data through a neural network. The algorithm provides two scoring functions based on energy and another based on the reconstruction error. For our experiments, we consider the energy-based anomaly scoring function, where samples with high energy are classified as anomalies.

**Deep Auto Encoding Gaussian Mixture Model [50].** DAGMM consists of two neural networks: an autoencoder and an estimation network. These networks are trained in an end-to-end fashion. The model concatenates the output of the encoder – the latent representation of the input – and the reconstruction error, then feeds them to the estimation network, whose output is used to compute the parameters of a Gaussian Mixture Model (GMM). Specifically, the estimation network is used to obtain sample likelihood, which is also considered the anomaly score.

**Adversarially Learned Anomaly Detection [48].** ALAD extends BiGAN [20] architecture by adding two more discriminators to ensure data-space and latent-space cycle consistencies. The scoring function is the reconstruction error on the output of an intermediate layer of one of the discriminators.

**Neural Transformation Learning for Deep Anomaly Detection Beyond Images [36].** NeuTraLAD is a self-supervised learning method for AD. It combines contrastive learning with the idea of learning data transformation through neural networks to do data augmentation with tabular data. The loss function is defined to maximize agreement between an input and its transformations while minimizing agreement between transformations of an input. The same deterministic loss function is used as the scoring function.

### 5.3 Results and Discussion

Table 2 reports F1-score, precision, and recall of the models with different values of contamination ratio  $\alpha$ .

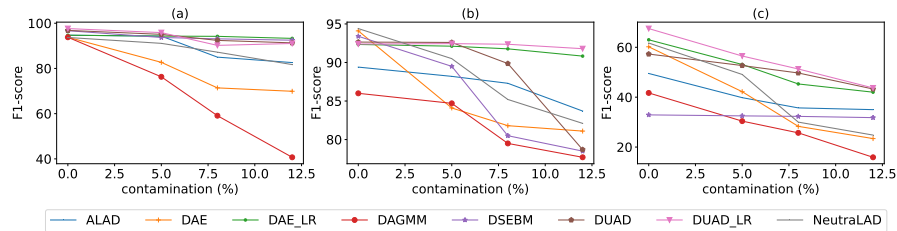


Fig. 6: Visualization models’ average F1-scores, with the training data contamination ratio on the x-axis. (a) For KDDCUP dataset. (b) For NSL-KDD dataset. (c) For CSE-CIC-IDS2018 dataset.

Table 2: Average Precision, Recall, and F1-Score (all with standard deviation) of the models trained exclusively on normal samples contaminated with a ratio  $\alpha$  of attacks samples.

Model	$\alpha$	KDDCUP 10			NSL-KDD			CSE-CIC-IDS2018		
		Pr	R	$F_1$	Pr	R	$F_1$	Pr	R	$F_1$
ALAD	0%	92.5 $\pm$ 0.9	97.2 $\pm$ 5.5	94.7 $\pm$ 3.2	87.1 $\pm$ 2.5	91.9 $\pm$ 1.1	89.4 $\pm$ 1.4	47.6 $\pm$ 8.4	52.0 $\pm$ 5.8	49.5 $\pm$ 6.6
DAGMM		91.2 $\pm$ 4.9	96.7 $\pm$ 3.0	93.8 $\pm$ 3.6	86.4 $\pm$ 4.2	85.9 $\pm$ 2.4	86.0 $\pm$ 1.8	42.4 $\pm$ 11.6	41.3 $\pm$ 10.1	41.7 $\pm$ 10.8
DSEBM		94.1 $\pm$ 0.3	99.4 $\pm$ 0.1	96.7 $\pm$ 0.1	93.6 $\pm$ 0.3	93.1 $\pm$ 0.4	93.4 $\pm$ 0.1	29.6 $\pm$ 0.1	37.0 $\pm$ 0.4	32.9 $\pm$ 0.2
NeuTraAD		88.2 $\pm$ 0.4	99.9 $\pm$ 0.1	93.7 $\pm$ 0.2	95.0 $\pm$ 0.9	93.8 $\pm$ 0.7	<b>94.4<math>\pm</math>0.3</b>	69.2 $\pm$ 4.0	55.7 $\pm$ 0.9	61.7 $\pm$ 1.4
DAE		88.8 $\pm$ 0.4	99.4 $\pm$ 0.4	93.8 $\pm$ 0.1	95.3 $\pm$ 1.7	93.2 $\pm$ 1.2	94.2 $\pm$ 0.3	61.9 $\pm$ 8.2	59.5 $\pm$ 5.6	60.2 $\pm$ 4.3
DUAD		94.3 $\pm$ 0.2	99.5 $\pm$ 0.0	96.8 $\pm$ 0.1	92.2 $\pm$ 0.2	93.2 $\pm$ 0.3	92.6 $\pm$ 0.1	56.5 $\pm$ 0.3	58.2 $\pm$ 0.1	57.3 $\pm$ 0.1
<b>DAE-LR</b>		90.0 $\pm$ 0.1	99.9 $\pm$ 0.1	94.7 $\pm$ 0.1	92.3 $\pm$ 0.1	92.5 $\pm$ 0.1	92.4 $\pm$ 0.0	69.5 $\pm$ 3.6	57.6 $\pm$ 0.8	63.0 $\pm$ 1.5
<b>DUAD-LR</b>		96.0 $\pm$ 0.2	99.3 $\pm$ 0.0	<b>97.6<math>\pm</math>0.1</b>	92.3 $\pm$ 0.1	92.5 $\pm$ 0.1	92.4 $\pm$ 0.0	62.2 $\pm$ 3.9	73.8 $\pm$ 0.6	<b>67.5<math>\pm</math>2.3</b>
ALAD	5%	90.6 $\pm$ 4.8	98.5 $\pm$ 1.0	94.3 $\pm$ 3.0	88.2 $\pm$ 2.3	88.2 $\pm$ 4.2	88.2 $\pm$ 2.8	36.9 $\pm$ 2.4	43.4 $\pm$ 5.4	39.8 $\pm$ 3.6
DAGMM		76.1 $\pm$ 21.0	77.3 $\pm$ 21.6	76.3 $\pm$ 20.9	83.0 $\pm$ 6.0	86.5 $\pm$ 4.6	84.7 $\pm$ 5.1	31.6 $\pm$ 14.2	29.9 $\pm$ 8.4	30.4 $\pm$ 11.0
DSEBM		88.1 $\pm$ 0.1	100.0 $\pm$ 0.0	93.7 $\pm$ 0.0	86.0 $\pm$ 1.6	93.2 $\pm$ 1.8	89.5 $\pm$ 1.7	29.2 $\pm$ 0.1	36.6 $\pm$ 0.1	32.5 $\pm$ 0.1
NeuTraAD		85.1 $\pm$ 0.5	98.0 $\pm$ 1.4	91.1 $\pm$ 0.7	87.0 $\pm$ 1.8	94.3 $\pm$ 1.9	90.5 $\pm$ 1.9	47.2 $\pm$ 5.4	51.9 $\pm$ 7.6	49.1 $\pm$ 5.5
DAE		79.6 $\pm$ 7.5	86.4 $\pm$ 10.2	82.7 $\pm$ 8.3	82.6 $\pm$ 2.6	85.5 $\pm$ 4.2	83.9 $\pm$ 1.1	39.5 $\pm$ 6.0	45.9 $\pm$ 8.6	42.2 $\pm$ 6.7
DUAD		90.7 $\pm$ 0.0	100.0 $\pm$ 0.0	95.1 $\pm$ 0.0	92.4 $\pm$ 0.3	92.9 $\pm$ 0.2	<b>92.6<math>\pm</math>0.0</b>	49.2 $\pm$ 0.3	56.9 $\pm$ 0.2	52.7 $\pm$ 0.2
<b>DAE-LR</b>		89.3 $\pm$ 0.1	100.0 $\pm$ 0.0	94.3 $\pm$ 0.1	92.1 $\pm$ 0.1	92.8 $\pm$ 0.1	92.4 $\pm$ 0.0	54.9 $\pm$ 5.2	52.0 $\pm$ 3.8	53.1 $\pm$ 2.2
<b>DUAD-LR</b>		92.2 $\pm$ 0.2	99.8 $\pm$ 0.0	<b>95.9<math>\pm</math>0.1</b>	92.5 $\pm$ 0.9	90.9 $\pm$ 1.1	91.7 $\pm$ 0.2	51.9 $\pm$ 5.2	62.1 $\pm$ 5.9	<b>56.5<math>\pm</math>5.4</b>
ALAD	8%	80.5 $\pm$ 9.6	90.6 $\pm$ 15.8	85.0 $\pm$ 12.4	87.3 $\pm$ 3.9	87.5 $\pm$ 2.2	87.3 $\pm$ 2.0	34.9 $\pm$ 7.1	36.8 $\pm$ 6.2	35.7 $\pm$ 6.1
DAGMM		60.4 $\pm$ 11.7	59.0 $\pm$ 16.8	59.1 $\pm$ 13.4	77.9 $\pm$ 9.5	81.4 $\pm$ 7.8	79.5 $\pm$ 8.3	23.7 $\pm$ 10.1	28.3 $\pm$ 12.9	25.7 $\pm$ 11.3
DSEBM		87.3 $\pm$ 0.2	99.6 $\pm$ 0.2	93.1 $\pm$ 0.2	77.3 $\pm$ 0.7	84.0 $\pm$ 0.8	80.5 $\pm$ 0.7	29.0 $\pm$ 0.1	36.5 $\pm$ 0.2	32.3 $\pm$ 0.1
NeuTraAD		81.5 $\pm$ 2.7	94.0 $\pm$ 2.2	87.2 $\pm$ 2.1	82.5 $\pm$ 1.7	88.1 $\pm$ 2.7	85.2 $\pm$ 1.6	27.3 $\pm$ 13.2	33.3 $\pm$ 16.2	30.0 $\pm$ 14.5
DAE		68.5 $\pm$ 3.8	75.1 $\pm$ 10.2	71.4 $\pm$ 5.9	80.2 $\pm$ 2.4	84.8 $\pm$ 2.8	82.4 $\pm$ 0.9	25.8 $\pm$ 8.6	31.6 $\pm$ 11.8	28.3 $\pm$ 9.9
DUAD		87.6 $\pm$ 0.4	97.8 $\pm$ 0.3	92.4 $\pm$ 0.3	88.0 $\pm$ 0.4	91.9 $\pm$ 0.7	89.9 $\pm$ 0.6	45.1 $\pm$ 0.3	55.3 $\pm$ 0.4	49.7 $\pm$ 0.3
<b>DAE-LR</b>		89.1 $\pm$ 0.1	100.0 $\pm$ 0.0	<b>94.2<math>\pm</math>0.0</b>	91.8 $\pm$ 0.1	93.0 $\pm$ 0.1	<b>92.4<math>\pm</math>0.0</b>	43.9 $\pm$ 2.9	47.2 $\pm$ 4.0	45.3 $\pm$ 2.5
<b>DUAD-LR</b>		83.7 $\pm$ 0.3	97.8 $\pm$ 0.1	90.2 $\pm$ 0.2	92.9 $\pm$ 1.3	90.3 $\pm$ 1.6	91.6 $\pm$ 0.2	46.7 $\pm$ 6.8	56.9 $\pm$ 8.0	<b>51.3<math>\pm</math>7.2</b>
ALAD	12%	77.6 $\pm$ 9.6	89.0 $\pm$ 17.3	82.6 $\pm$ 13.2	80.6 $\pm$ 4.3	87.0 $\pm$ 4.5	83.7 $\pm$ 4.4	33.0 $\pm$ 4.0	37.5 $\pm$ 4.4	35.0 $\pm$ 3.7
DAGMM		42.8 $\pm$ 13.5	39.3 $\pm$ 14.9	40.7 $\pm$ 13.9	75.9 $\pm$ 14.0	79.7 $\pm$ 13.5	77.7 $\pm$ 13.6	13.4 $\pm$ 7.8	20.6 $\pm$ 12.7	15.9 $\pm$ 9.0
DSEBM		86.7 $\pm$ 0.4	99.1 $\pm$ 0.3	92.5 $\pm$ 0.4	77.5 $\pm$ 0.4	79.5 $\pm$ 0.2	78.5 $\pm$ 0.2	28.6 $\pm$ 0.1	35.7 $\pm$ 0.2	31.8 $\pm$ 0.2
NeuTraAD		75.4 $\pm$ 3.2	89.4 $\pm$ 2.8	81.7 $\pm$ 2.5	81.2 $\pm$ 2.7	83.2 $\pm$ 3.0	82.1 $\pm$ 0.9	24.2 $\pm$ 11.8	25.8 $\pm$ 14.2	24.8 $\pm$ 12.8
DAE		69.0 $\pm$ 7.2	71.4 $\pm$ 8.5	69.9 $\pm$ 6.2	79.7 $\pm$ 1.8	83.1 $\pm$ 1.8	81.3 $\pm$ 0.5	21.0 $\pm$ 5.9	26.6 $\pm$ 7.6	23.4 $\pm$ 6.5
DUAD		85.9 $\pm$ 0.2	97.5 $\pm$ 0.3	91.3 $\pm$ 0.2	77.1 $\pm$ 0.2	80.4 $\pm$ 0.1	78.7 $\pm$ 0.1	41.3 $\pm$ 0.6	45.8 $\pm$ 1.1	43.3 $\pm$ 0.8
<b>DAE-LR</b>		88.3 $\pm$ 0.4	98.9 $\pm$ 0.2	<b>93.3<math>\pm</math>0.2</b>	90.8 $\pm$ 0.3	92.9 $\pm$ 0.2	<b>91.8<math>\pm</math>0.1</b>	43.2 $\pm$ 3.8	41.6 $\pm$ 5.3	42.0 $\pm$ 2.2
<b>DUAD-LR</b>		85.4 $\pm$ 0.4	97.6 $\pm$ 0.2	91.0 $\pm$ 0.4	90.8 $\pm$ 1.0	92.2 $\pm$ 0.6	91.5 $\pm$ 0.4	41.2 $\pm$ 0.9	46.9 $\pm$ 0.2	<b>43.7<math>\pm</math>0.1</b>

**Performance on KDDCUP dataset.** When there is no contamination, all models perform well, with an average F1-score above 90% across 20 runs. DSEBM and DUAD-LR outperform the other models with an F1-score greater than 96%. From the two-dimensional representation of a subset of the KDDCUP dataset in Figure 5a, it is clear that normal and attack data are mostly distinguishable, with minimal overlap. This may explain the good performance of all models on the KDDCUP dataset. Figure 6a shows performance degradation for all models as the contamination ratio increases, albeit to varying extents. Notably, the F1-scores of DAGMM and DAE decrease dramatically from 93.8% to 40.7% and from 93.8% to 69.9%, respectively, with a contamination ratio of 12%. Remarkably, DAE-LR and DUAD-LR maintain good performance even under varying levels of contamination.

DAGMM estimates a Gaussian mixture without explicitly handling outliers during training, causing contaminated samples to influence density estimation and shift normal samples toward low-density regions. However, DUAD and DSEBM show resistance to contamination. DUAD’s approach of re-evaluating the training data seems to eliminate some contaminated samples, allowing for training on a nearly clean subset. Even with 12% contamination, DUAD-LR and DAE-LR demonstrate significant improvements over their standard counterparts.

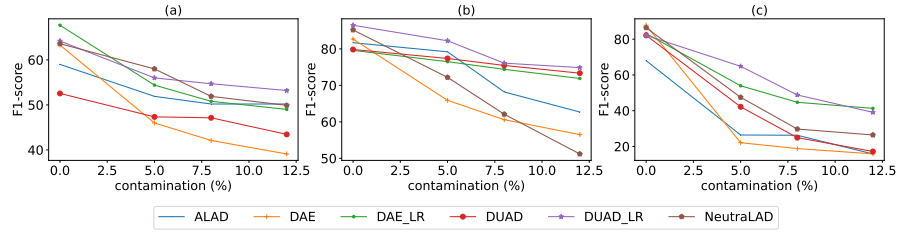


Fig. 7: Models’ average F1-scores, with the training data contamination ratio on the x-axis. (a) For Kitsune dataset, (b) For CIC-IoT23 dataset, (c) For the revised version of CIC-IDS 2018 dataset.

As for DSEBM, recall remains around 99%, while precision decreases. This suggests that DSEBM effectively classifies all attacks, assigning high energy to them, regardless of the contamination level. However, DSEBM’s precision decreases as contamination in the training data increases."

**Performance on NSL-KDD dataset.** Figure 6b presents different patterns than those observed with the KDDCUP dataset. Note that NSL-KDD is a relatively balanced dataset. For instance, DSEBM shows poor performance, with its F1-score dropping from 93% to 78% at a contamination rate of 12%. This may be attributed to the removal of duplicates in the KDDCUP dataset. In contrast, DUAD-LR’s distribution clustering and latent regularizer play a central role in defending against contamination of the training set. DUAD consistently performs well, maintaining an F1-score of 91%, regardless of the contamination level. Furthermore, DAE-LR outperformed the other models when faced with 12% contamination.

**Performance on Kitsune(IoT) and CIC-IoT23.** Based on the ranking in Figure 8, we have selected the top models, namely ALAD, DAE, DAE-LR, DUAD, DUAD-LR, and NeutraLAD which we trained on Kitsune and CIC-IoT23 datasets. Figure 7 displays their performance under varying levels of data contamination. We observe that NeutraLAD outperforms the others with clean training data by the f1-score of 85.2% on CIC-IoT23. However, its performance suffered a significant drop of over 30% when trained with a 12% contamination rate. This emphasizes the challenge of this self-supervised approach in handling data contamination. On the other hand, DAE-LR and DUAD-LR still better perform with different levels of contamination in both Kitsune and CIC-IoT23. We also note that none of the models was able to achieve an F1-score exceeding 70% on the Kitsune dataset. This poor performance reveals how hard it is to detect advanced cyber attacks on this dataset in an unsupervised fashion.

**Performance on CSE-CIC-IDS2018 dataset.** On this dataset, DUAD-LR achieves the highest F1-score (67%), followed by DAE-LR (63%) and NeutraLAD (61%). Overall, all the models struggle to achieve high performance with regards to what they yielded on KDDCUP and NSL-KDD datasets, even without contamination. In Figure 5c, we present a t-SNE visualization of a subset

of normal data and infiltration attack data from the CSE-CIC-IDS2018 dataset. Interestingly, we notice a significant number of overlaps between benign traffic and attack, rendering their distinction challenging. In Figure 6c, a decrease in F1-scores for all models is observed as the contamination rate increases. While DUAD-LR has previously demonstrated resilience to contamination, its F1-score decreases from 67% to 43% on CSE-CIC-IDS2018 with a contamination rate of 12%. This decline in performance can be attributed to the fact that, unlike KDDCUP and NSL-KDD (as illustrated in Figures 5a and 5b), some of the attack samples in CSE-CIC-IDS2018 (Figure 5c) exhibit less dissimilarity from normal traffic data. Consequently, eliminating contamination from the training set, even for DUAD-LR, proves to be challenging. Nonetheless, DUAD-LR consistently maintains a higher F1-score compared to other models across different contamination rates, followed by DAE-LR and then ALAD. A similar performance trend is observed on the revised version of this dataset. However, the models exhibit superior performance in the absence of contamination, followed by a steep decline as the contamination ratio increases, as shown in Figure 7c.

### Discussion

This study covers a wide range of approaches in unsupervised AD, showing that generative models like ALAD and self-supervised models like NeuTralAD can achieve great performance in AD tasks, including network intrusion detection. The overall ranking of the models is illustrated in Figure 8. The bold

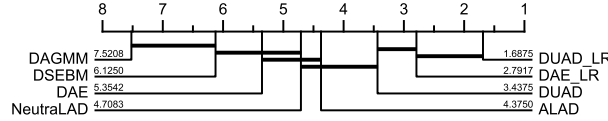


Fig. 8: CD-Diagram for overall average rank based on F1-scores of the models on all datasets with different levels of contamination. The lower the ranking, the better the model performs compared to others.

horizontal lines connecting groups of models indicate that they are not statistically different based on  $p$ -values. DUAD-LR, DAE-LR, ALAD, and NeuTralAD emerge as the top-performing models overall. Particularly in scenarios where the training set is contaminated, DUAD-LR and DAE-LR stand out, demonstrating the benefit of regulating the latent representation for AD tasks involving contaminated datasets. Moreover, results obtained on KDDCUP and NSL-KDD may be misleading when compared to the realities of current computer networks. Models that excel on these outdated datasets may not perform as well on more relevant datasets such as CSE-CIC-IDS2018, Kitsune, and CIC-IoT23, which better simulate the current state of computer networks and cyber threats. Thus, there is a pressing need for more representative network intrusion detection datasets.

**Future directions.** Based on our experiments, it becomes apparent that data contamination can have a negative impact on the performance of even the most advanced models. Given the unpredictable nature of data cleanliness in real-world scenarios, it is primordial to incorporate defenses against contamination when developing DL models for cybersecurity. One potential defense strategy could involve inferring data labels during the training process. Once inferred labels are available, the algorithm could dynamically identify and remove potentially anomalous instances from the training dataset, enhancing robustness.

We observed that DUAD-LR exhibits consistent resilience to contamination due to its approach of re-evaluating the training set through clustering. However, one drawback of DUAD is its relatively slow training process, especially on large datasets, as it performs clustering over the entire training set multiple times. To address this, one could explore extending the clustering concept introduced by DUAD to identify and reject contaminated data in an online and more computationally efficient manner.

## 6 Conclusion

In this rapidly evolving era of cyberattacks, data-driven anomaly detection methods in cybersecurity must demonstrate resilience against data contamination attacks. This paper has contributed valuable insights into the robustness of recent data-driven anomaly detection models when applied to network intrusion detection datasets. Our experiments demonstrate that the latent regulated auto-encoder that we propose in this study, leads to competitive performance when the training data is contaminated. We also observe that even state-of-the-art deep learning models for AD experience a notable drop in performance when the training data is contaminated by attack instances. This underscores the critical significance of robustness to contamination as a pivotal criterion in selecting an anomaly detection model for cybersecurity applications. Furthermore, our study underscores the potential pitfalls of relying solely on outdated cybersecurity datasets for model evaluation. The experiments conducted reveal that model performance on such datasets may lead to misguided conclusions when applied to the ever-evolving landscape of computer networks and emerging attack patterns. In conclusion, our findings provide insights that will inform ongoing efforts in the development of cyberdefense solutions. This includes the imperative need for designing defense frameworks that address training set contamination challenges for deep anomaly detection methods in the context of cybersecurity. These insights are pivotal in enhancing the resilience and effectiveness of anomaly detection systems in the presence of persistent cyberattacks.

## References

1. Alvarez, M., Verdier, J.C., Nkashama, D.K., Frappier, M., Tardif, P.M., Kabanza, F.: A revealing large-scale evaluation of unsupervised anomaly detection algorithms. arXiv preprint arXiv:2204.09825 (2022)



2. Awan, S., Luo, B., Li, F.: Contra: Defending against poisoning attacks in federated learning. In: Computer Security–ESORICS 2021: 26th European Symposium on Research in Computer Security, Darmstadt, Germany, October 4–8, 2021, Proceedings, Part I 26. pp. 455–475. Springer (2021)
3. Biggio, B., Nelson, B., Laskov, P.: Poisoning attacks against support vector machines. arXiv preprint arXiv:1206.6389 (2012)
4. Biggio, B., Roli, F.: Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition* **84**, 317–331 (2018)
5. Bouthillier, X., Delaunay, P., Bronzi, M., Trofimov, A., Nichyporuk, B., Szeto, J., Mohammadi Sepahvand, N., Raff, E., Madan, K., Voleti, V., et al.: Accounting for variance in machine learning benchmarks. *Proceedings of Machine Learning and Systems* **3**, 747–769 (2021)
6. Bovenzi, G., Aceto, G., Ciunzio, D., Montieri, A., Persico, V., Pescapé, A.: Network anomaly detection methods in iot environments via deep learning: A fair comparison of performance and robustness. *Computers & Security* **128** (2023)
7. Bovenzi, G., Foggia, A., Santella, S., Testa, A., Persico, V.: Data poisoning attacks against autoencoder-based anomaly detection models: a robustness analysis. In: IEEE International Conference on Communications. pp. 5427–5432 (2022)
8. Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J.: Lof: identifying density-based local outliers. In: Proceedings of the 2000 ACM SIGMOD international conference on Management of data. pp. 93–104 (2000)
9. Chalapathy, R., Chawla, S.: Deep learning for anomaly detection: A survey. arXiv preprint arXiv:1901.03407 (2019)
10. Chen, Z., Yeo, C.K., Lee, B.S., Lau, C.T.: Autoencoder-based network anomaly detection. In: Wireless Telecommunications Symposium (WTS). pp. 1–5 (2018)
11. Chicco, D., Jurman, G.: The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC genomics* **21**(1), 1–13 (2020)
12. Crawford, M., Khoshgoftaar, T.M., Prusa, J.D., Richter, A.N., Al Najada, H.: Survey of review spam detection using machine learning techniques. *Journal of Big Data* **2**(1), 1–24 (2015)
13. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine learning research* **7**, 1–30 (2006)
14. Dua, S., Du, X.: Data mining and machine learning in cybersecurity. CRC press (2016)
15. Fang, M., Cao, X., Jia, J., Gong, N.: Local model poisoning attacks to {Byzantine-Robust} federated learning. In: 29th USENIX Security Symposium (USENIX Security 20). pp. 1605–1622 (2020)
16. Ferencz, K., Domokos, J., Kovacs, L.: Review of industry 4.0 security challenges. In: 2021 IEEE 15th international symposium on applied computational intelligence and informatics (SACI). pp. 245–248 (2021)
17. Gail, M.S., Richard, F.: Using Effect Size—or Why the P Value Is Not Enough. *Journal of Graduate Medical Education* **4**(3), 279–282 (2012)
18. Gong, D., Liu, L., Le, V., Saha, B., Mansour, M.R., Venkatesh, S., Hengel, A.v.d.: 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)
19. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. *Advances in neural information processing systems* **27** (2014)

20. Gupta, P., Tyagi, P., Singh, R.K.: Analysis of generative adversarial networks for data-driven inverse airfoil design. In: *Proceedings of International Conference on Information Technology and Applications*. pp. 251–261. Springer (2022)
21. Hayase, J., Kong, W., Somani, R., Oh, S.: Spectre: Defending against backdoor attacks using robust statistics. *arXiv preprint arXiv:2104.11315* (2021)
22. Ilyas, A., Santurkar, S., Tsipras, D., Engstrom, L., Tran, B., Madry, A.: Adversarial examples are not bugs, they are features. *Advances in neural information processing systems* **32** (2019)
23. Jagielski, M., Oprea, A., Biggio, B., Liu, C., Nita-Rotaru, C., Li, B.: Manipulating machine learning: Poisoning attacks and countermeasures for regression learning. In: *2018 IEEE Symposium on Security and Privacy (SP)*. pp. 19–35 (2018)
24. Kwon, D., Kim, H., Kim, J., Suh, S.C., Kim, I., Kim, K.J.: A survey of deep learning-based network anomaly detection. *Cluster Computing* **22**(1), 949–961 (2019)
25. Kye, H., Kim, M., Kwon, M.: Hierarchical detection of network anomalies : A self-supervised learning approach. *IEEE Signal Processing Letters* **29**, 1908–1912 (2022). <https://doi.org/10.1109/LSP.2022.3203296>
26. Li, T., Wang, Z., Liu, S., Lin, W.Y.: Deep unsupervised anomaly detection. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. pp. 3636–3645 (2021)
27. Liu, F.T., Ting, K.M., Zhou, Z.H.: Isolation forest. In: *2008 eighth ieee international conference on data mining*. pp. 413–422. IEEE (2008)
28. Liu, L., Engelen, G., Lynar, T., Essam, D., Joosen, W.: Error prevalence in nids datasets: A case study on cic-ids-2017 and cse-cic-ids-2018. In: *2022 IEEE Conference on Communications and Network Security (CNS)*. pp. 254–262 (2022)
29. Liu, Z., Thapa, N., Shaver, A., Roy, K., Yuan, X., Khorsandroo, S.: Anomaly detection on iot network intrusion using machine learning. In: *2020 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD)*. pp. 1–5. IEEE (2020)
30. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. *Journal of machine learning research* **9**(11) (2008)
31. Madani, P., Vlajic, N.: Robustness of deep autoencoder in intrusion detection under adversarial contamination. In: *Proceedings of the 5th Annual Symposium and Bootcamp on Hot Topics in the Science of Security. HoTSoS '18, Association for Computing Machinery, New York, NY, USA* (2018)
32. Muñoz-González, L., Biggio, B., Demontis, A., Paudice, A., Wongrassamee, V., Lupu, E.C., Roli, F.: Towards poisoning of deep learning algorithms with back-gradient optimization. In: *Proceedings of the 10th ACM workshop on artificial intelligence and security*. pp. 27–38 (2017)
33. Nelson, B., Barreno, M., Chi, F.J., Joseph, A.D., Rubinstein, B.I., Saini, U., Sutton, C., Tygar, J.D., Xia, K.: Exploiting machine learning to subvert your spam filter. *LEET* **8**(1), 9 (2008)
34. Neto, E.C.P., Dadkhah, S., Ferreira, R., Zohourian, A., Lu, R., Ghorbani, A.A.: Ciciot2023: A real-time dataset and benchmark for large-scale attacks in iot environment. *Sensors* **23**(13) (2023)
35. Nkashama, D., Soltani, A., Verdier, J.C., Frappier, M., Tardif, P.M., Kabanza, F.: Robustness evaluation of deep unsupervised learning algorithms for intrusion detection systems. *ICML workshop on Machine Learning for Cybersecurity (ICML-ML4Cyber)* (2022)

36. Qiu, C., Pfrommer, T., Kloft, M., Mandt, S., Rudolph, M.: Neural transformation learning for deep anomaly detection beyond images. In: International Conference on Machine Learning. pp. 8703–8714. PMLR (2021)
37. Rubinstein, B.I., Nelson, B., Huang, L., Joseph, A.D., Lau, S.h., Rao, S., Taft, N., Tygar, J.D.: Antidote: understanding and defending against poisoning of anomaly detectors. In: Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement. pp. 1–14 (2009)
38. Ruff, L., Kauffmann, J.R., Vandermeulen, R.A., Montavon, G., Samek, W., Kloft, M., Dietterich, T.G., Müller, K.R.: A unifying review of deep and shallow anomaly detection. *Proceedings of the IEEE* (2021)
39. Ruff, L., Vandermeulen, R., Goernitz, N., Deecke, L., Siddiqui, S.A., Binder, A., Müller, E., Kloft, M.: Deep one-class classification. In: International conference on machine learning. pp. 4393–4402. PMLR (2018)
40. Saito, T., Rehmsmeier, M.: The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PloS one* **10**(3), e0118432 (2015)
41. Sarker, I.H., Khan, A.I., Abushark, Y.B., Alsolami, F.: Internet of things (iot) security intelligence: a comprehensive overview, machine learning solutions and research directions. *Mobile Networks and Applications* **28**(1), 296–312 (2023)
42. Schölkopf, B., Williamson, R.C., Smola, A., Shawe-Taylor, J., Platt, J.: Support vector method for novelty detection. *Advances in neural information processing systems* **12** (1999)
43. Sharafaldin, I., Lashkari, A.H., Ghorbani, A.A.: Toward generating a new intrusion detection dataset and intrusion traffic characterization. (2018)
44. Simmross-Wattenberg, F., Asensio-Perez, J.I., Casaseca-De-La-Higuera, P., Martin-Fernandez, M., Dimitriadis, I.A., Alberola-Lopez, C.: Anomaly detection in network traffic based on statistical inference and  $\alpha$ -stable modeling. *IEEE Transactions on Dependable and Secure Computing* **8**(4), 494–509 (2011)
45. Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.A.: 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)
46. Wang, B., Cao, X., Gong, N.Z., et al.: On certifying robustness against backdoor attacks via randomized smoothing. *arXiv preprint arXiv:2002.11750* (2020)
47. Xiao, H., Biggio, B., Brown, G., Fumera, G., Eckert, C., Roli, F.: Is feature selection secure against training data poisoning? In: international conference on machine learning. pp. 1689–1698. PMLR (2015)
48. Zenati, H., Romain, M., Foo, C.S., Lecouat, B., Chandrasekhar, V.: Adversarially learned anomaly detection. In: 2018 IEEE International conference on data mining (ICDM). pp. 727–736. IEEE (2018)
49. Zhai, S., Cheng, Y., Lu, W., Zhang, Z.: Deep structured energy based models for anomaly detection. In: International conference on machine learning. pp. 1100–1109. PMLR (2016)
50. Zong, B., Song, Q., Min, M.R., Cheng, W., Lumezanu, C., Cho, D., Chen, H.: Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In: International conference on learning representations (2018)