A Robust Prioritized Anomaly Detection when Not All Anomalies are of Primary Interest

Guanyu Lu[1], Fang Zhou[1]*, Martin Pavlovski[2], Chenyi Zhou[1], Cheqing Jin[1]
[1]School of Data Science and Engineering, East China Normal University, China [2]Temple University, PA, USA
[1]{gylu, cyzhou_2000}@stu.ecnu.edu.cn, {fzhou, cqjin}@dase.ecnu.edu.cn, [2]martin.pavlovski@temple.edu

***Abstract*—Anomaly detection has emerged as a prominent research area with extensive exploration across various applications. Existing methods predominantly focus on detecting all anomalies exhibiting unusual patterns, however, they overlook the critical need to prioritize the detection of target anomaly categories (anomalies of primary interest) that could pose significant threats to various systems. This oversight results in the excessive involvement of valuable human labor and resources in dealing with non-target anomalies (that are of lower interest). This work is focused on target-class anomaly detection, which entails overcoming several challenges: (1) deficient prior information regarding non-target anomalies and (2) an elevated false positive rate caused by the presence of non-target anomalies. Thus, we introduce a novel semi-supervised model, called TargAD, which leverages a few labeled target anomalies, along with potential non-target anomaly candidates and normal candidates selected from unlabeled data. By introducing a novel loss function, TargAD effectively maximizes the distributional differences among normal candidates, target anomalies, and non-target anomaly candidates, leading to a significant improvement in detecting target anomalies. Furthermore, when confronted with novel non-target anomaly scenarios, TargAD maintains its accuracy in detecting target anomalies. We conducted extensive experiments, the results of which demonstrate that TargAD outperforms eleven state-of-the-art baselines on a real-world dataset and three publicly available datasets, with average AUPRC improvements of 5.9%-24.8%, 9.2%-57.8%, 2.7%-71.3%, and 2.0%-70.3%, respectively.**

I. Introduction

The anomaly detection problem aims to identify instances in data that deviate significantly from normal patterns [1], and it finds broad applications across various domains, including financial fraud detection [2], network intrusion detection [3], healthcare disease detection [4], and image tampering detection [5], among others. Given the scarcity of anomalies and the challenges associated with gathering extensive labeled data, unsupervised anomaly detection methods have emerged as dominant solutions [6]–[8]. These methods leverage underlying properties of the data, such as distance and density measures, to effectively detect anomalies. Nevertheless, in practice, typically a small number of labeled anomalies is available. Therefore, semi-supervised anomaly detection methods [9] utilize easily accessible labeled data that provide valuable prior knowledge for learning representations of normal/anomalous instances [10]–[14], thus significantly improving detection accuracy compared to unsupervised methods.

*Corresponding author.

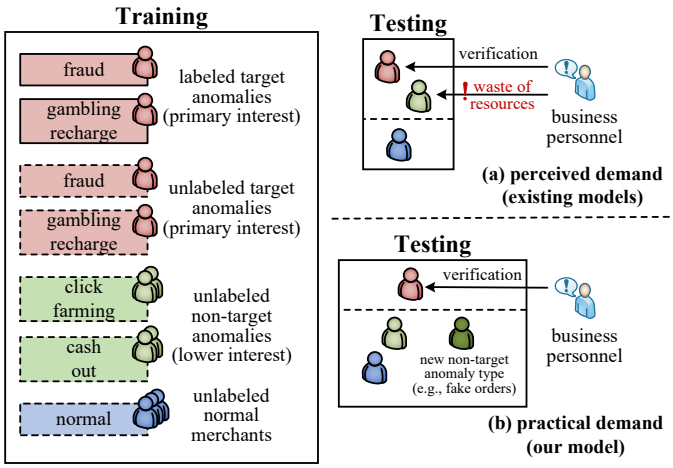


Fig. 1. Perceived demand against practical demand for anomaly detection within an integrated payment platform. Training data includes labeled target anomalies, as well as unlabeled target and non-target anomalies and normal merchants. (a) Perceived demand (top). Testing data includes unlabeled normal merchants, target anomalies, and non-target anomalies. The goal is to identify any anomalies that deviate from usual patterns. (b) Practical demand (bottom). Testing data may also contain new types of non-target anomalies. The focus shifts to identifying only target anomalies of primary interest.

The existing methods have demonstrated decent performance; however, they assume a uniform risk level for all types of anomalies and assign equal priority to their identification. This assumption does not align with the practical demands inherent in diverse real-world scenarios. We use the following two cases to illustrate this issue.

- Consider an anomaly detection scenario within an integrated payment platform (as illustrated in Fig. 1), where over six million merchants are involved. A small subset of these merchants exhibits anomalous behaviors, categorized into two distinct risk levels: high-risk anomalies that could lead to severe economic losses (e.g., fraud and gambling recharge), and low-risk anomalies that present minimal threats (e.g., click farming and cash out). While high-risk anomalies are relatively scarce, typically amounting to only *a few dozen* per day, the quantity of low-risk anomalies significantly surpasses that of high-risk anomalies, reaching into the *thousands*. The ratio of low-risk to high-risk anomalies ranges from 20 to 60 times. Therefore, if all risk levels of anomalies are to be verified, a substantial amount of human labor and time (from days up to several weeks) would be required.

- Another example is found in the operation of a large-scale enterprise network, which generates hundreds of millions of data packets daily. These data packets potentially contain various network attacks that could pose threats to computer systems, but the risk level of these attacks varies significantly. Some high-risk anomalous events, such as distributed denial-of-service (DDoS) attacks and advanced persistent threats (APT), may occur with a low probability yet have the potential to bring down an entire system [15], [16]. Conversely, low-risk anomalies like spam and SQL injection attempts occur nearly every day but typically do not present actual threats, rendering the implementation of corresponding security measures unnecessary [17]. Expecting enterprise security personnel to oversee all risk levels of anomalies would place a heavy burden on their shoulders.

Given the varying levels of risks associated with anomalies, and taking into account the economic, human, and time costs involved, it becomes imperative to prioritize the identification of anomalies and specifically target the group of high-risk anomalies that cause severe harm. These anomalies of high interest, referred to as *target anomalies*, represent the primary focus of detection. The remaining groups of anomalies can be regarded as *non-target anomalies* due to their negligible impact. This requirement motivates us to delve into the following research question: **Is it feasible to develop an approach capable of precisely identifying target anomalies, considering that not all anomalies are of primary interest?**

We now present two primary challenges encountered when addressing this research question: (1) *The scarcity of prior information on non-target anomalies.* Despite the larger quantity of non-target anomalies compared to target anomalies, the number of types of non-target anomalies is significantly greater than that of target anomalies. Since non-target anomalies pose minimal threats, it would be easier to disregard them. Furthermore, new types of non-target anomalies may occasionally occur, in which case it would be challenging to label all types of non-target anomalies in time. (2) *High false positive rate induced by the presence of non-target anomalies.* Non-target anomalies, in fact, exhibit "abnormal" characteristics compared to normal instances. The existing approaches for anomaly detection, whether unsupervised [18] or semi-supervised [11], [13], [19], [20], do not prioritize the identification of target anomalies. Due to the considerable quantity of non-target anomalies, the existing approaches may tend to identify non-target anomalies, thus leading to misidentification of target anomalies that carry greater significance.

To address the aforementioned challenges, we propose a novel model called TargAD[1] (Target-class Anomaly Detection), which leverages a small number of labeled target anomalies, as well as potential normal and non-target anomaly candidates chosen from unlabeled data, to better detect anomalies that are of primary interest. To address the issue of lack of priors on non-target anomalies, we utilize multiple Auto-Encoders (AEs) to segregate unlabeled data into normal candidates and non-target anomaly candidates based on AEs' reconstruction errors. A novel loss function is introduced with the intention of maximizing the distributional disparities among normal candidates, target anomalies, and non-target anomaly candidates. Inspired by the concept of outlier exposure [21], we take non-target anomaly candidates as out-of-distribution instances compared to normal instances and target anomalies and calibrate the predictive distribution of non-target anomalies toward a uniform distribution. This mechanism enables the proposed model to effectively identify target anomalies even when new types of non-target anomalies are introduced.

TargAD offers several advantages in anomaly detection. First, this study is focused on the detection of target anomalies, and TargAD provides an effective solution to address this research problem. Moreover, in contrast to prevailing anomaly detection methods that classify all types of anomalies into a single class, TargAD possesses an additional advantage: the ability to identify non-target anomalies as a separate group when the need arises to identify other types of anomalies (refer to Section III-C). These advantages provide a flexible strategy for real-world applications. The prompt detection of target anomalies, which carry higher significance, enables timely attention and facilitates the implementation of corresponding actions. When sufficient time and human resources are available, attention can then be given to the remaining types of anomalies, namely non-target anomalies.

To validate the effectiveness of TargAD in identifying target anomalies, we conducted experiments on a real-world dataset and three publicly available datasets commonly used for anomaly detection. We compared TargAD with eleven state-of-the-art anomaly detection methods, and the results demonstrate that TargAD exhibits outstanding performance in terms of AUROC and AUPRC metrics. Particularly, it excels in terms of AUPRC, showcasing average improvements of 5.9%-24.8%, 9.2%-57.8%, 2.7%-71.3%, and 2.0%-70.3% across the four datasets, respectively. In addition, extensive experiments were conducted to evaluate the model's robustness, where TargAD emerged as the top performer as well.

In brief, the main contributions of this paper are:

- We discuss an innovative anomaly detection scenario where only the anomalies of interest (often wreaking havoc) need detection. To the best of our knowledge, this work represents the first endeavor to accurately identify target anomalies.
- We present a model named TargAD to tackle the challenges encountered in this scenario. A novel loss function is introduced to maximize the distributional discrepancies among normal candidates, target anomalies, and non-target anomaly candidates. Additionally, we devise a weight-updating mechanism to effectively mitigate the presence of noise among non-target anomaly candidates.
- TargAD exhibits an additional advantage that sets it apart from other baselines, enabling it to effectively differentiate between normal instances, target anomalies, and non-

[1]The code is available at https://github.com/ZhouF-ECNU/TargAD.

target anomalies. In essence, TargAD primarily focuses on detecting target anomalies while also possessing the ability to identify non-target anomalies.

II. Related work

In this section, we present the related work within anomaly detection and out-of-distribution (OOD) detection. The pivotal objective of our work is to detect target anomalies, and we draw into OOD ideas to decrease false positives triggered by non-target anomalies.

Anomaly Detection. Anomaly detection refers to the procedure of identifying data instances that are inconsistent with the majority of instance patterns. Owing to the difficulty of collecting large amounts of labeled data, anomaly detection approaches have primarily focused on unsupervised and semi-supervised learning perspectives. Unsupervised perspectives typically include isolation-based [18], density-based [22], distance-based [23], probability-based [24], and reconstruction-based [6] methods, but such methods lack consideration of prior knowledge regarding real anomalies and tend to exhibit high false positive rates on real-world datasets. In contrast, semi-supervised methods take advantage of a small portion of labeled data for training, avoiding this problem. Our model likewise utilizes a few labeled anomalies, so we mainly introduce semi-supervised anomaly detection methods below. Positive-unlabeled (PU) learning methods [25]–[27] are extensively applied in semi-supervised anomaly detection, leveraging positive instances and unlabeled data, but such methods assume that anomalies are homogeneous. In fact, anomalies do not satisfy a necessarily unified pattern. Anomaly detection approaches based on Generative Adversarial Networks (GANs) [28], [29] detect anomalies by quantifying the difference between real and generated instances as an anomaly score. Some of these methods also incorporate a limited number of labeled anomalies into their process. For example, PIA-WAL [29] leverages a small set of labeled anomalies to guide adversarial learning and generates peripheral normal instances through a weighted generative model to solve the problem of insufficient learning of such instances. Several distance-based [11], [12] semi-supervised anomaly detection methods have demonstrated remarkable performance. DeepSAD [11] is an improvement upon unsupervised DeepSVDD [23] by leveraging labeled anomalies, as its loss function penalizes the inverse of the distance of these anomalies to the latent space's center such that they map farther from the center. Note that the anomaly detection problem addressed in this work differs from error detection (e.g., Raha [30]). Error detection aims to identify features' values that deviate from the established ground truth, whereas our study focuses on detecting instance anomalies.

In real-world scenarios, there are certain anomalies that are not the primary interest, also known as non-target anomalies. This poses a significant challenge for semi-supervised models as they struggle to identify target anomalies effectively. In contrast to these methods, TargAD takes a different approach inspired by the concept of OOD. It ensures that the prediction probability distribution of non-target anomalies tends to a uniform distribution, mitigating the challenges associated with misidentifying non-target anomalies as target anomalies.

Out-of-distribution Detection. The task of OOD detection involves identifying test instances that originate from a distribution with semantic deviations from the in-distribution (ID). For instance, in the application of autonomous driving [31], the system may encounter anomalies during operation that were not observed during the training phase. *On the one side*, experts and scholars actively explore methods to effectively characterize the distributional disparities between instances from ID and OOD categories. OOD detection using deep models dates back to a baseline [32], which utilizes the maximum softmax probability as the score for ID-ness. However, overfitting of deep neural networks often results in high confidence (i.e., softmax overconfidence problem) for OOD instances, prompting extensive research to address this challenge [33]–[36]. In a breakthrough, certain studies [37], [38] suggest utilizing an energy function instead of softmax for OOD detection to alleviate the overconfidence problem. *On the flip side*, some researchers gather an auxiliary set of OOD instances to assist the model in learning ID/OOD discrepancy, i.e., Outlier Exposure (OE) [21]. However, the space of collected OOD data may contain numerous uninformative instances; thus, recent studies have attempted to tackle this problem by rejecting fuzzy instances around boundaries [39] and mining valuable outliers [40], [41]. The traditional OE model typically exhibits awful detection performance when confronted with unseen OOD instances in the testing data. Consequently, DOE [42] implicitly leads to data conversion through model perturbation to ensure consistency in the distribution of diverse unseen OOD cases.

Unlike OOD detection approaches, our focus is not on detecting such OOD cases but rather on avoiding the misidentification of non-target anomalies as target anomalies by utilizing the concept of outlier exposure. In addition, impure auxiliary OOD data (mixed with ID instances) leads to poor performance of the aforementioned OE models. For this challenge, we develop a novel weighting mechanism in the TargAD framework that assigns higher weights to non-target anomalies and dynamically updates the weights with each iteration.

III. Methodology

A. Problem Definition

Let $\mathcal{D}$ be a training set such that $\mathcal{D} = \mathcal{D}_L \cup \mathcal{D}_U$ ($|\mathcal{D}_L| \ll |\mathcal{D}_U|$). Namely, $\mathcal{D}_L = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_r, y_r)\}$ denotes a set of labeled target anomalies. Each $\boldsymbol{x} \in \mathcal{D}_L$ is a D-dimensional instance associated with a target anomaly type $y \in \{1, \ldots, m\}$, where m is the number of target anomaly types. On the other hand, $\mathcal{D}_U = \{\boldsymbol{x}_{r+1}, \ldots, \boldsymbol{x}_N\}$ represents an unlabeled dataset comprising numerous normal instances, a fraction of non-target anomalies, and a fraction of target anomalies. Note that, in practical applications, normal instances in $\mathcal{D}_U$ may form several groups. Hence, we use k to denote the number of such hidden groups.

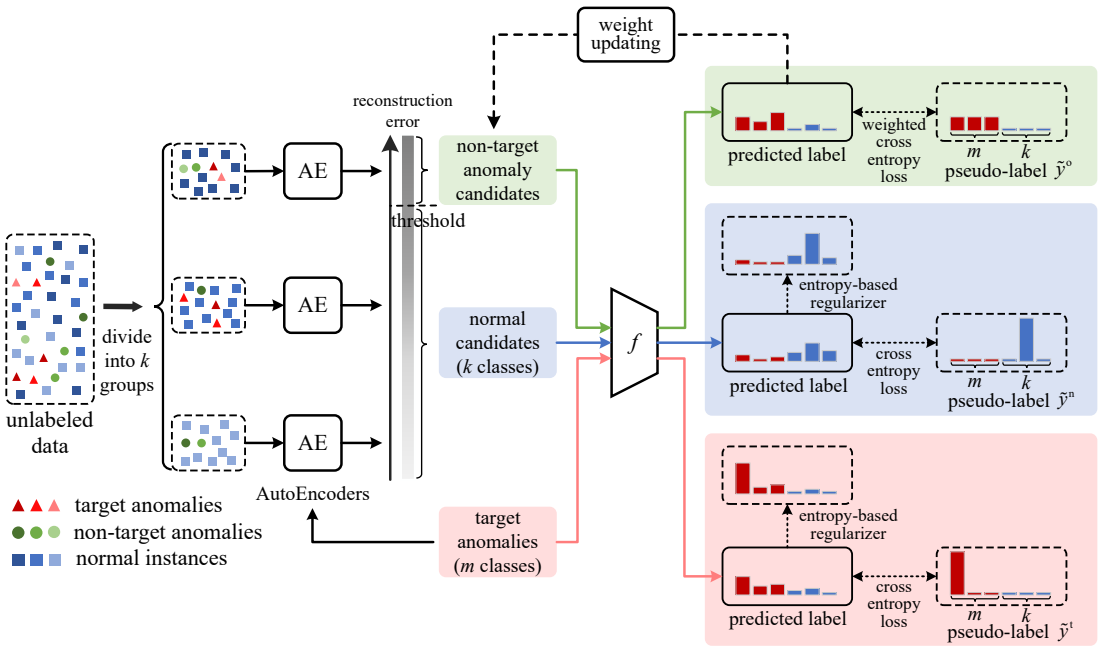


Fig. 2. The workflow of TargAD.

The goal is to learn a classifier f that predicts whether a given instance $\boldsymbol{x}$ is a target anomaly (anomaly of primary interest) or not. The designated output label for target anomalies is $+1$, while -1 is assigned to normal instances and non-target anomalies.

B. Proposed Approach

In this section, we introduce the proposed TargAD model. Fig. 2 provides an overview of the model's workflow, encompassing candidate selection and detection components. As there is a lack of prior knowledge about non-target anomalies and normal instances, we first apply clustering and autoencoders to select normal and non-target anomaly candidates. We next propose a novel loss function in the detection component to maximize distribution differences among target anomalies, non-target anomaly candidates, and normal candidates and further detect target anomalies precisely.

1) Candidate Selection: The objective of this component is to surmount the challenge arising from the absence of prior knowledge regarding non-target anomalies and normal instances. It involves the selection of non-target anomaly candidates and normal candidates from the unlabeled data so as to provide essential information on non-target anomalies and normal instances for the subsequent detection phase.

Note that $\mathcal{D}_U$ contains numerous normal instances with diverse patterns. For example, in the context of detecting fraud in credit card transactions, normal individuals could be categorized into groups based on their spending habits, such as low-consumption and high-consumption groups, which exhibit significant variations in transaction amounts. Thus, we first apply k-means clustering to partition the unlabeled data into k groups, denoted as $\{\mathcal{D}_{U_1}, \mathcal{D}_{U_2}, \ldots, \mathcal{D}_{U_k}\}$, ensuring that the subsequent autoencoders will individually learn from each group of instances to capture a more precise normal pattern for the corresponding group.

Considering that autoencoders have demonstrated robust feature learning capabilities coupled with relatively low complexity, we apply an autoencoder to each cluster $\mathcal{D}_{U_i}$ $(i \in \{1, 2, \ldots, k\})$ to separate normal and anomalous instances. The traditional autoencoder is based solely on unsupervised learning. However, since a few labeled target anomalies are available in our setting, inspired by DeepSAD [11], we leverage such labeled anomalies to modify the loss function of the traditional autoencoder. More precisely, the objective of the i^{th} autoencoder AE_i is to minimize the following loss:

$$\begin{aligned}\mathcal{L}_{\text{AE}_i} = &\frac{1}{|\mathcal{D}_{U_i}|}\sum_{\boldsymbol{x}\in\mathcal{D}_{U_i}} \left\|\boldsymbol{x} - \phi_i^D\left(\phi_i^E(\boldsymbol{x})\right)\right\|^2 \\ &+ \frac{\eta}{|\mathcal{D}_L|} * \sum_{\boldsymbol{x}\in\mathcal{D}_L} \left(\left\|\boldsymbol{x} - \phi_i^D\left(\phi_i^E(\boldsymbol{x})\right)\right\|^2\right)^{-1},\end{aligned} \tag{1}$$

where ϕ_i^E and ϕ_i^D are the encoder and decoder networks of the i^{th} autoencoder AE_i, respectively, and η is the trade-off parameter. The first term of $\mathcal{L}_{\text{AE}_i}$ is used to minimize the reconstruction error on unlabeled data. The second term penalizes the inverse of the reconstruction error for labeled anomalies, encoding the normal instances in the low-dimensional space in a more compact manner. As the majority of instances in $\mathcal{D}_U$ are normal, they are easier to reconstruct from the low-dimensional space of the autoencoder.

After completing the training of the k autoencoders in parallel, for an unlabeled instance $\boldsymbol{x} \in \mathcal{D}_{U_i}$, we calculate its reconstruction error, i.e.,

$$S^{\text{Rec}}(\boldsymbol{x}) = \left\|\boldsymbol{x} - \phi_i^D\left(\phi_i^E(\boldsymbol{x})\right)\right\|^2. \tag{2}$$

A higher reconstruction error indicates a greater likelihood of the instance being anomalous. We sort the unlabeled data in descending order based on their reconstruction errors. The reconstruction error of the instance ranked at $\alpha\%$ is selected as the threshold, that is, the top $\alpha\%$ of instances are considered as non-target anomaly candidates, denoted as $\mathcal{D}_U^A$. The remaining data, which can be more easily reconstructed from low-dimensional spaces, are regarded as normal candidates, denoted as $\mathcal{D}_U^N$.

2) Detection: The objective of this component is to address the issue of false positives caused by misidentifying non-target as target anomalies. More precisely: (1) We assign different forms of pseudo-labels to distinguish between labeled target anomalies, normal candidates, and non-target anomaly candidates. Through the pseudo-labels assigned to non-target anomaly candidates, we calibrate the predictive distribution of non-target anomalies toward a uniform distribution. One advantage of the proposed pseudo-label design is that it ensures the predictive probability distribution is calibrated to a uniform distribution when encountering new types of non-target anomalies. (2) We maximize the distributional differences between normal candidates, target anomalies, and non-target anomaly candidates by introducing a novel loss function. (3) We effectively tackle noise (comprising target anomalies and inaccurately reconstructed normal instances) among the non-target anomaly candidates. For this purpose, we devise a weight-updating mechanism that assigns higher weights to potential non-target anomalies.

We commence by describing the pseudo-label setup. Assume that the category information of the labeled anomalies is already provided and involves m classes. The class labels of the normal candidates are assigned based on their corresponding clustering indices. Such category information is utilized as pseudo-labels encoded into one-hot vectors, guiding the subsequent classification. Specifically, for anomalies, 1 is assigned in one of the first m dimensions of the pseudo-label, such as $\tilde{y}^{\mathrm{t}} = \{\underbrace{0,1,0,\cdots,0}_{m},\underbrace{0,\cdots,0}_{k}\}$. For normal candidates, 1 is assigned in one of the last k dimensions of the pseudo-label, such as $\tilde{y}^{\mathrm{n}} = \{\underbrace{0,\cdots,0}_{m},\underbrace{0,\cdots,0,1,0}_{k}\}$.

To learn the distinction between normal and anomalous patterns, we train a conventional classifier, denoted as f (a simple and effective multi-layer perceptron network is utilized in our work), on $\mathcal{D}_U^N$ and $\mathcal{D}_L$ using the standard cross-entropy loss, i.e.,

$$\begin{aligned}\mathcal{L}_{\mathrm{CE}} =& \frac{1}{|\mathcal{D}_L|}\sum_{\boldsymbol{x}\in\mathcal{D}_L}\sum_{j=1}^{m+k}-\tilde{y}_j^{\mathrm{t}} * \log\left(p_j(\boldsymbol{x})\right) \\ &+\frac{1}{|\mathcal{D}_U^N|}\sum_{\boldsymbol{x}\in\mathcal{D}_U^N}\sum_{j=1}^{m+k}-\tilde{y}_j^{\mathrm{n}} * \log\left(p_j(\boldsymbol{x})\right),\end{aligned} \tag{3}$$

where $p_j(\boldsymbol{x})$ represents the probability that classifier f predicts that $\boldsymbol{x}$ belongs to the j^{th} class, and $\tilde{y}_j^{\mathrm{t}}$ and $\tilde{y}_j^{\mathrm{n}}$ are the j^{th} elements of pseudo-labels $\tilde{y}^{\mathrm{t}}$ and $\tilde{y}^{\mathrm{n}}$, respectively. The objective of $\mathcal{L}_{\mathrm{CE}}$ is to correctly classify target anomalies and normal instances.

We then introduce the design of pseudo-labels for non-target anomalies. Inspired by OE [21], we treat non-target anomalies as out-of-distribution instances since they are unseen anomalies. According to OE [21], the pseudo-labels of non-target anomalies are initially set to $\underbrace{\left\{\frac{1}{m+k},\cdots,\frac{1}{m+k},\cdots,\frac{1}{m+k}\right\}}_{m+k}$ to satisfy a uniform distribution. However, this pseudo-label setting fails to adequately emphasize the anomalous nature of non-target anomalies. Therefore, we adjust the pseudo-label setting to $\tilde{y}^{\mathrm{o}} = \{\underbrace{\frac{1}{m},\cdots,\frac{1}{m}}_{m},\underbrace{0,\cdots,0}_{k}\}$ to inform the classifier that non-target anomalies are different from normal instances and do not belong to any known types of target anomalies. This modification makes the model more effective at distinguishing between normal and anomalous patterns. Besides, in the event of emerging new types of non-target anomalies, since their learned representation patterns differ from those of target anomalies and normal instances, the predictive distributions of these novel types of instances are calibrated to be uniform.

Since non-target anomalies exist in $\mathcal{D}_U^A$, as such were obtained during the candidate selection phase, we assign $\tilde{y}^{\mathrm{o}} = \{\underbrace{\frac{1}{m},\cdots,\frac{1}{m}}_{m},\underbrace{0,\cdots,0}_{k}\}$ to the instances in $\mathcal{D}_U^A$. However, non-target anomaly candidates may contain target anomalies and even some inaccurately reconstructed normal instances (erroneously filtered as non-target anomaly candidates during candidate selection). The modified pseudo-label could uniform the prediction probabilities of these instances. To address this issue, we propose a novel weight-updating strategy that aims to mitigate the impact of noisy instances in non-target anomaly candidates. We utilize the maximum predicted probabilities across all dimensions of an instance $\boldsymbol{x}$ to calculate its weight,

$$w(\boldsymbol{x}) = \frac{\max_{\boldsymbol{x}'\in\mathcal{D}_U^A}\epsilon(\boldsymbol{x}') - \epsilon(\boldsymbol{x})}{\max_{\boldsymbol{x}'\in\mathcal{D}_U^A}\epsilon(\boldsymbol{x}') - \min_{\boldsymbol{x}'\in\mathcal{D}_U^A}\epsilon(\boldsymbol{x}')}, \tag{4}$$

such that $\epsilon(\boldsymbol{x}) = \max_{j=1}^{j=m+k} p_j(\boldsymbol{x})$. Owing to the crafted pseudo-labeling mechanism, when dealing with non-target anomaly candidates, the $\epsilon(\boldsymbol{x})$ values of normal instances and target anomalies generally exhibit larger values than those of non-target anomalies. Consequently, non-target anomalies in $\mathcal{D}_U^A$ can be assigned larger weights according to Eq. (4). The initial weights are assigned as

$$w_0(\boldsymbol{x}) = \frac{\max_{\boldsymbol{x}'\in\mathcal{D}_U^A} S^{\mathrm{Rec}}(\boldsymbol{x}') - S^{\mathrm{Rec}}(\boldsymbol{x})}{\max_{\boldsymbol{x}'\in\mathcal{D}_U^A} S^{\mathrm{Rec}}(\boldsymbol{x}') - \min_{\boldsymbol{x}'\in\mathcal{D}_U^A} S^{\mathrm{Rec}}(\boldsymbol{x}')}, \tag{5}$$

where $S^{\mathrm{Rec}}(\boldsymbol{x})$ is the reconstruction error of $\boldsymbol{x}$ in candidate selection. Since the reconstruction errors of normal instances are smaller than those of both target and non-target anomalies, normal instances are assigned larger weights based on Eq. (5) during the initialization phase. After applying Eq. (4), the weights assigned to non-target anomalies undergo an increment (see Fig. 5(a)).

We design an improved loss that incorporates modified pseudo-labels and a weight parameter w to constrain the non-target anomaly candidates, i.e.,

$$\mathcal{L}_{\mathrm{OE}} = \frac{1}{\left|\mathcal{D}_U^A\right|} \sum_{\boldsymbol{x} \in \mathcal{D}_U^A} w(\boldsymbol{x}) * \sum_{j=1}^{m+k} -\tilde{y}_j^{\mathrm{o}} * \log\left(p_j(\boldsymbol{x})\right), \quad (6)$$

where $\tilde{y}_j^{\mathrm{o}}$ is the j^{th} element of the pseudo-label $\tilde{y}^{\mathrm{o}}$. The tailored loss encourages non-target anomalies to exhibit uniform distributions across the first m dimensions.

During the initial training iterations, the classifier has not yet gained adequate knowledge about target anomalies and normal instances. Influenced by Eq. (6), the probabilities of target anomalies and normal instances are learned so as to follow a certain degree of uniform distribution, which consequently engenders low confidence in predicting target anomalies and normal instances, as they may exist among non-target anomaly candidates. To address this concern, we utilize the entropy of the outputs generated by f on $\mathcal{D}_U^N$ and $\mathcal{D}_L$ as a regularizer to boost the confidence of the classifier's predictions:

$$\mathcal{L}_{\mathrm{RE}} = \frac{1}{\left|\mathcal{D}_L\right| + \left|\mathcal{D}_U^N\right|} \sum_{\boldsymbol{x} \in \left(\mathcal{D}_L \cup \mathcal{D}_U^N\right)} \sum_{j=1}^{m+k} p_j(\boldsymbol{x}) * \log\left(p_j(\boldsymbol{x})\right). \quad (7)$$

The classifier is optimized by jointly minimizing the three afore discussed loss functions:

$$\mathcal{L}_{\mathrm{clf}} = \mathcal{L}_{\mathrm{CE}} + \lambda_1 * \mathcal{L}_{\mathrm{OE}} + \lambda_2 * \mathcal{L}_{\mathrm{RE}}, \quad (8)$$

where λ_1 and λ_2 are trade-off parameters.

3) Training and Inference: Algorithm 1 outlines the training procedure of TargAD. The training data is comprised of a few labeled target anomalies and a large number of unlabeled instances. As the unlabeled instances consist mainly of normal instances with multiple patterns, the first step is to apply k-means to cluster the unlabeled instances into different groups (Line 1). Next, the labeled target anomalies and each unlabeled instance cluster are used to train the corresponding autoencoder, with reconstruction errors serving as selection scores for the unlabeled instances (Lines 2-5). All unlabeled instances are sorted in descending order of their reconstruction errors, and the top $\alpha\%$ instances are selected to form a non-target anomaly candidate set, while the remaining instances form a normal candidate set (Lines 6-7). Instances from these two sets, along with labeled anomalies, are then fed into a classifier for classification. A novel loss function is designed to maximize differences in distribution between normal candidates, different types of target anomalies, and non-target anomaly candidates (Lines 8-16). The training process culminates in a trained classifier.

For the testing data, a probability distribution is obtained for each instance using the softmax function of the trained classifier f. An anomaly score of an instance is calculated based on the maximum softmax probability among the first m dimensions of the probability distribution, that is,

$$S^{\mathrm{tar}}(\boldsymbol{x}) = \max_{j \in \{1,2,.,m\}} p_j(\boldsymbol{x}). \quad (9)$$

Algorithm 1: Training TargAD

input : Training set $\mathcal{D}$, clustering hyperparameter k, threshold $\alpha\%$, trade-off parameters η, λ_1, λ_2, number of classifier's iterations $epochs$

output: A trained classifier f (a multi-layer perceptron network)

1 Apply k-means on $\mathcal{D}_U$ and then group $\mathcal{D}_U$ into k clusters denoted as $\{\mathcal{D}_{U_1}, \mathcal{D}_{U_2}, \ldots, \mathcal{D}_{U_k}\}$;
2 **for** $i \leftarrow 1$ **to** k **do**
3 Train the corresponding AE_i on $\mathcal{D}_{U_i}$ using Eq. (1);
4 Calculate the reconstruction error $S^{\mathrm{Rec}}(\boldsymbol{x})$ for each instance in $\mathcal{D}_{U_i}$ using Eq. (2);
5 **end**
6 Sort all instances in $\mathcal{D}_U$ in descending order of $S^{\mathrm{Rec}}(\boldsymbol{x})$;
7 Select top $\alpha\%$ instances as $\mathcal{D}_U^A$ (non-target anomaly candidates) and remaining instances as $\mathcal{D}_U^N$;
8 Randomly initialize the classifier f's parameters;
9 **for** $i \leftarrow 1$ **to** $epochs$ **do**
10 **if** $i = 1$ **then**
11 Initialize the weights of non-target anomaly candidates using Eq. (5);
12 **else**
13 Update the weights of non-target anomaly candidates using Eq. (4);
14 **end**
15 Update the parameters of f using Eq. (8);
16 **end**
17 Return f.

Thus, a greater value indicates a higher likelihood of the instance being the target anomaly.

4) Model Complexity: In the following, we discuss the time complexity of TargAD. The datasets utilized are all in tabular format, with the size of the input data $\mathcal{D}$ represented as $N \times D$. We first analyze the complexity of candidate selection (Lines 1-7 in Algorithm 1). The unlabeled data $\mathcal{D}_U$ undergoes clustering into k groups which, assuming a maximum of t iterations of clustering, has a complexity of $O\left(t \times k \times N \times D\right)$. Given that t and k can be treated as constants, and that typically $t, k \ll N$ or/and $t, k \ll D$ (depending on the dataset characteristics), the complexity can be simplified to $O\left(ND\right)$. The data from the k clusters are input into the corresponding autoencoders for parallel training. The autoencoders used to select candidates are bottleneck networks consisting of two structurally symmetrical multi-layer perceptron networks. Each network comprises L hidden layers, with the l-th hidden layer having d_l neurons, and the representation dimension of the last hidden layer being d. The time complexity of the feed-forward computation of the parallel-trained autoencoders in each epoch is $O\left(\frac{N}{k} \times 2 \times \left(Dd_1 + \sum_{j=1}^{L-1} d_j d_{j+1} + d_l d\right)\right)$ at best (when each cluster is assigned an equal number of instances) and $O\left(N \times 2 \times \left(Dd_1 + \sum_{j=1}^{L-1} d_j d_{j+1} + d_l d\right)\right)$ at worst (when all instances are assigned to one cluster).

TABLE I
DETAILED STATISTICS OF THE UNSW-NB15, KDDCUP99, NSL-KDD, AND SQB DATASETS.

datasets	D^*	training set		validation set			testing set		
		labeled target	unlabeled	normal	target	non-target	normal	target	non-target
UNSW-NB15	196	300	62,631	14,899	334	450	18,601	1,666	2,335
KDDCUP99	32	200	58,524	13,918	419	188	17,380	799	352
NSL-KDD	41	200	45,385	10,743	487	366	13,492	749	629
SQB	182	212	132,028	14,671**	23	142	148,323**	236	1,502

* D is the data dimensionality.
** Since the pure normal instance set cannot be obtained in the SQB dataset, and considering the substantial presence of normal instances within the unlabeled data, we treat the unlabeled data as normal instances for verification and testing.

Due to the relatively small size of the hidden layer units and representation dimensions compared to the input data dimensions, the complexity of training the autoencoders can be expressed as $O(ND)$. In addition, the process of ranking the reconstruction error scores involves a time complexity of $O(N\log N)$, thus the overall complexity of candidate selection can be expressed as $O(ND + N\log N)$.

The classifier can be conceptualized as a multi-layer perceptron network, with $L^{'}$ hidden layers, $d^{'}_{l'}$ number of neurons in the $l^{'}$-th hidden layer, and $d^{'}$ denoting the representation dimension of the last hidden layer. That being said, the time complexity of training the classifier (Lines 8-17 in Algorithm 1) is $O\left(N \times \left(Dd^{'}_1 + \sum_{j=1}^{L^{'}-1} d^{'}_j d^{'}_{j+1} + d^{'}_{l'} d^{'}\right)\right)$, and it can likewise be further simplified to $O(ND)$, which is linear concerning both the input data volume and the input data dimension.

C. TargAD's Additional Advantage

The objective of the proposed approach is to identify target anomalies effectively. The prediction is performed via a binary classifier based on Eq. (9). Essentially, the utilization of distinct pseudo-label settings for normal candidates, target anomalies, and non-target anomaly candidates confers an additional advantage to TargAD. It enables TargAD to effectively identify these three categories of instances based on the model's output, setting itself apart from other alternatives in the domain. Specifically, an instance is classified as normal if the sum of the predicted probabilities in the last k dimensions is larger than $\frac{k}{m+k}$, that is, $\sum_{j=m+1}^{m+k} p_j(\boldsymbol{x}) > \frac{k}{m+k}$. Otherwise, the instance is considered anomalous. Notice that, in this work, non-target anomalies are taken as out-of-distribution instances due to their unseen labels. Thus, we can segregate non-target anomalies into a distinct group by applying out-of-distribution detection strategies (such as Maximum Softmax Probability (MSP) [32], Energy Score (ES) [37], and Energy Discrepancy (ED) [43]). Experiments in Section IV-D6 provide an extensive discussion of these three strategies to assess their effectiveness in distinguishing normal instances, target anomalies, and non-target anomalies.

IV. EXPERIMENTS

A. Datasets

Due to our work assuming the existence of target and non-target anomaly classes, for experimentation it is essential to consider datasets that include multiple anomaly classes, rather than using data containing a single type of anomaly, as some previous works [12], [13], [18] have done. With this assumption in mind, three publicly available datasets (UNSW-NB15[2], KDDCUP99[3], and NSL-KDD[4]) that are widely utilized in the field of anomaly detection and one real-world dataset were selected to evaluate the effectiveness of our model. Detailed statistics for the training, validation, and testing sets of these four datasets can be found in Table I.

The UNSW-NB15 dataset comprises seven distinct anomaly classes, with each instance in the dataset being characterized by 196 features. Three anomaly types, namely *Generic*, *Backdoor*, and *DoS*, were designated as target anomaly classes, while the *Fuzzers*, *Analysis*, *Exploits*, and *Reconnaissance* classes were considered as non-target anomaly classes. The KDDCUP99 dataset originally comprises numerous redundant features, and we retained 32 features for each instance by eliminating redundancy. We selected the *R2L* and *DoS* classes as the target anomaly classes and the *probe* class as the non-target anomaly class. The NSL-KDD dataset serves as a revised version of the KDDCUP99 dataset, the instances which are described by 41 features. In this dataset, we retained the same target and non-target anomaly classes as those present in the KDDCUP99 dataset. For these three datasets, we conducted random sampling to obtain a limited number of labeled data regarding the target class anomalies. These labeled anomalies accounted for approximately 0.16% to 0.48% of the training data. This aligns with a real anomaly detection scenario, where a vast majority of the data is unlabeled, and only a tiny portion is labeled. We selected a portion of target and non-target anomalies and integrated them with normal instances to create an unlabeled training dataset, with a default contamination rate of 5%.

We also utilized the SQB dataset, a real-world dataset, which comprises the daily transactions of merchants on an integrated payment platform[5] in China. Its primary objective is to identify merchants engaged in target anomalous activities that cause serious harm, such as fraud and gambling recharge. Conversely, anomalies with relatively lower risk levels, such as click farming and cash out, are considered non-target anomaly

[2]https://research.unsw.edu.au/projects/unsw-nb15-dataset
[3]https://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
[4]https://www.unb.ca/cic/datasets/nsl.html
[5]https://www.shouqianba.com/

classes. Using transaction data collected from the integrated payment platform spanning April 2021 to April 2022, we extracted a total of 182 features, including transaction amount, transaction frequency, payment type, etc. We constructed a dataset comprising 295,022 unlabeled instances, 471 target anomalies, and 1,644 non-target anomalies. It is worth noting that the unlabeled data within the SQB dataset conceals numerous instances of both target and non-target anomalies, but the exact proportion of contamination remains unknown.

We preprocessed all four of the utilized datasets, applied one-hot encoding to the categorical features (where applicable), and mapped all features to the range of $[0, 1]$ using min-max normalization.

B. Baselines

We compare the performance of TargAD against eleven state-of-the-art anomaly detection methods. Among these methods, iForest [18] and REPEN [12] are unsupervised models, while the others are semi/weakly-supervised approaches.

- **iForest** [18] identifies anomalies in light of the number of times required to isolate instances by isolation trees.
- **REPEN** [12] is an instantiation of the RAMODO framework that learns a small set of features tailored for distance-based anomaly detectors.
- **ADOA** [19] clusters observed anomalies and applies filtering of unlabeled data, using the resulting instances to construct a weighted multi-class model.
- **FEAWAD** [44] encodes hidden representation, reconstruction residual vector, and reconstruction error into representations of input data for anomaly detection.
- **PUMAD** [25] uses a distance hashing-based method to filter unlabeled instances for deep metric learning.
- **DevNet** [13] employs end-to-end learning of anomaly scores based on neural deviation learning, leveraging a few labeled anomalies and Gaussian priors.
- **DeepSAD** [11] introduces a novel loss term for labeled anomalies, pushing them away from the one-class center.
- **DPLAN** [20], based on the deep reinforcement learning, is designed to empower anomaly detection agents to learn from labeled anomalies and extend the acquired anomaly patterns for detecting unknown anomalies.
- **PIA-WAL** [29] utilizes labeled anomalies to guide an adversarial training process and generates peripheral normal instances through a weighted generative model, aiming to better understand the characteristics of such instances.
- **Dual-MGAN** [45] combines two sub-GANs to perform active learning and data augmentation strategies to ensure accurate anomaly detection.
- **PReNet** [46] randomly samples two instances from the training set to form instance pairs and learns instance pair features and anomaly scores by predicting the relation of sampled instances.

C. Parameters Setup and Metrics

TargAD applied the Adaptive Moment Estimation (Adam) optimizer to update the model parameters. It was trained using a learning rate of 0.0001 and batches of 256 instances for $\mathcal{L}_{\mathrm{AE}_i}$, while for $\mathcal{L}_{\mathrm{clf}}$, a learning rate of 0.00001 and batches of 128 instances were used. The optimization process for both loss functions involved a total of 30 iterations each. The value of the clustering hyperparameter k was selected based on the elbow method. The threshold for candidate selection was set to 5%. The trade-off parameter η was set to 1 in the autoencoders' loss function. For the loss function $\mathcal{L}_{\mathrm{clf}}$, the trade-off parameters λ_1 and λ_2 were set to 0.1 and 1, respectively. These hyperparameters were determined based on the model's performance on a separate validation set. The initial parameters for the baselines were set according to the specifications provided in their respective papers and subsequently fine-tuned to achieve the best possible performance. All models were implemented in Python 3, and the experiments were conducted on an Alibaba Cloud DSW server equipped with Intel Xeon Platinum 8269CY CPU, Ubuntu 18.04 operating system, and 60 GB of memory.

We utilize two widely accepted and complementary performance metrics, Area Under the Precision Recall Curve (AUPRC) and Area Under the Receiver Operating Characteristic Curve (AUROC), to evaluate the effectiveness of TargAD and that of the baseline methods. In anomaly detection applications, AUPRC is considered a more suitable metric than AUROC as AUPRC reflects the identification performance of positive classes. A higher AUPRC value, closer to 1, indicates a more accurate detection of anomalous instances. The reported AUROC and AUPRC results for all models are the average values obtained from 5 independent runs. In addition, when analyzing the identification performance of non-target anomalies (see Section IV-D6), we also assessed additional metrics derived from the confusion matrix, including Precision, Recall, and F1-Score.

D. Experimental Results and Analysis

We address the following *research questions (**RQs**)* in the experimental analysis. The first five questions are related to detecting target anomalies, while the last question concerns the additional advantage that TargAD offers. ***RQ1:*** How does our model's performance on the four datasets compare to that of the baselines? ***RQ2:*** What is the performance and robustness of TargAD when confronted with situations involving (1) new types of non-target anomalies in the testing data, (2) variations within the number of target anomaly classes, (3) reductions of the amount of labeled target anomalies, and (4) fluctuations of the anomaly contamination rate? ***RQ3:*** What are the contributions of various terms in the loss function of the classifier? ***RQ4:*** Is the weight-updating strategy devised in $\mathcal{L}_{\mathrm{OE}}$ effective? ***RQ5:*** How sensitive is TargAD to the hyperparameter α, and trade-off parameters η, λ_1, and λ_2? ***RQ6:*** How effective is TargAD in discerning among normal instances, target anomalies, and non-target anomalies?

*1) Overall Performance (**RQ1**):* Table II presents the results of TargAD and the baselines on the four datasets with respect to the AUPRC and AUROC metrics. Our model demonstrates the most outstanding performance in terms of

TABLE II
AUROC AND AUPRC PERFORMANCE (± STANDARD DEVIATION) OF TARGAD AND ELEVEN SOTA BASELINE METHODS.

Models	AUPRC				AUROC			
	UNSW-NB15	KDDCUP99	NSL-KDD	SQB	UNSW-NB15	KDDCUP99	NSL-KDD	SQB
iForest	0.301±0.036	0.333±0.033	0.356±0.010	0.035±0.009	0.783±0.011	0.944±0.006	0.917±0.002	0.912±0.003
REPEN	0.276±0.039	0.545±0.016	0.524±0.078	0.013±0.001	0.875±0.016	0.957±0.006	0.905±0.009	0.855±0.003
ADOA	0.226±0.007	0.236±0.010	0.210±0.006	0.018±0.002	0.852±0.006	0.933±0.004	0.900±0.005	0.921±0.003
FEAWAD	0.540±0.031	0.593±0.033	0.741±0.013	0.057±0.036	0.946±0.010	0.975±0.008	0.968±0.005	0.942±0.012
PUMAD	0.573±0.011	0.922±0.027	0.691±0.039	0.202±0.017	0.903±0.023	0.982±0.003	0.954±0.018	0.978±0.006
DevNet	0.671±0.014	0.912±0.017	0.850±0.013	0.126±0.006	0.950±0.004	0.993±0.001	0.985±0.001	0.977±0.002
DeepSAD	0.677±0.017	0.765±0.018	0.752±0.029	0.132±0.003	0.974±0.001	0.993±0.001	0.986±0.001	**0.985**±0.001
DPLAN	0.658±0.037	0.834±0.066	0.832±0.029	0.151±0.005	0.951±0.006	0.985±0.004	0.973±0.007	0.971±0.003
PIA-WAL	0.698±0.024	0.780±0.074	0.893±0.010	0.139±0.010	0.946±0.010	0.977±0.007	0.981±0.001	0.963±0.005
Dual-MGAN	0.646±0.027	0.866±0.006	0.725±0.010	0.096±0.007	0.913±0.004	0.988±0.002	0.969±0.003	0.969±0.006
PReNet	0.712±0.009	0.920±0.003	0.787±0.022	0.125±0.002	0.937±0.001	0.992±0.001	0.983±0.001	0.972±0.002
TargAD	**0.804**±0.001	**0.949**±0.006	**0.913**±0.006	**0.261**±0.024	**0.978**±0.001	**0.994**±0.001	**0.988**±0.001	0.958±0.006

AUPRC and AUROC on the three publicly available datasets. It is evident that TargAD exhibits substantial advancements in accurately detecting target anomalies when compared to other baseline methods. TargAD consistently outperforms the baselines on the UNSW-NB15, KDDCUP99, and NSL-KDD datasets, showcasing average AUROC improvements of 0.4%-19.5%, 0.1%-6.1%, and 0.2%-8.8%, respectively. AUPRC serves as a reliable metric for identifying target anomalies when the data is unbalanced, and TargAD's AUPRC performance demonstrates average improvements of 9.2%-57.8%, 2.7%-71.3%, and 2.0%-70.3%, respectively. Further, TargAD's performance is considerably stable across the three publicly available datasets, as evidenced by its significantly smaller standard deviations of AUROC and AUPRC in comparison to the other methods. In the real-world application involving merchant anomaly detection (the SQB dataset), TargAD exhibits significant AUPRC performance improvements, with an average improvement ranging between 5.9% and 24.8%.

From Table II, we discerned that, except for ADOA, the performance of the eight semi-supervised methods surpasses that of the unsupervised iForest and REPEN. This discrepancy can be ascribed to the capability of the semi-supervised methods to effectively leverage the available supervision information obtained through labeled target anomalies. Despite constituting a tiny fraction of the training data (i.e., 0.16%-0.48%), these labeled instances play an indispensable role in bolstering the model's overall performance.

The next involves convergence analysis, which aims to assess the model's generalization performance and stability. Fig. 3(a) depicts the curve representing the loss values output by TargAD after each training epoch. Starting from the 15$^{\text{th}}$ epoch, the fluctuation in TargAD's loss value remains within a narrow range, indicating that the model gradually converges. Additionally, Fig. 3(b) presents the AUPRC performance at each epoch for TargAD and the semi-supervised baselines during the testing phase. Compared to other models, TargAD not only achieves the optimal AUPRC performance but also accomplishes this outcome with fewer epochs, highlighting the faster convergence of our model.

*2) Robustness Analysis (**RQ2**):* We conducted experiments on the UNSW-NB15 dataset to compare and analyze the

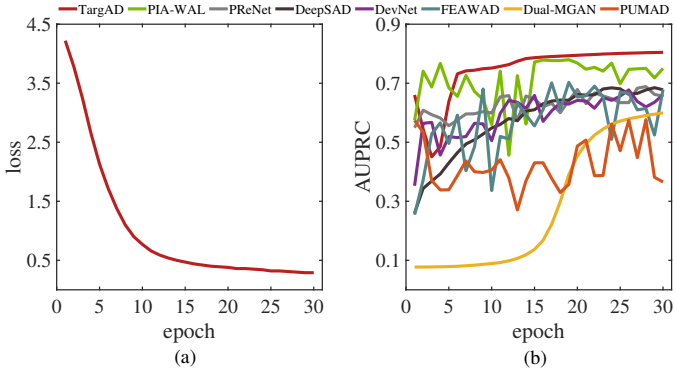


Fig. 3. Convergence analysis of the model. (a) The loss value of TargAD at each epoch during training. (b) AUPRC performance at each epoch for TargAD and baseline models during testing.

robustness of TargAD in detecting target anomalies against other semi-supervised baselines. This dataset was chosen as it encompasses multiple anomaly types, making it suitable for exploring various scenarios discussed in this section.

We first evaluated the performance of TargAD and other baselines in the presence of new types of non-target anomalies in the testing data. The testing set contains four specific non-target anomaly types (*Fuzzers*, *Analysis*, *Exploits*, and *Reconnaissance*). To investigate this scenario, we perturbed the number of non-target anomaly types among the unlabeled portion of the training data, varying this number among four different settings: 4 classes (four types of non-target anomalies identical to the ones originally present in the testing set), 3 classes (*Fuzzers*, *Analysis*, and *Reconnaissance*), 2 classes (*Analysis* and *Reconnaissance*), or 1 class (*Reconnaissance*). The objective is to determine the presence of 0, 1, 2, or 3 new types of non-target anomalies within the testing data. Fig. 4(a) shows the AUPRC performance of TargAD and other baseline methods in the context of varying numbers of novel non-target anomaly types in the testing set. TargAD exhibits better stability in detecting target anomalies when confronted with new non-target anomaly types, consistently maintaining an AUPRC performance of approximately 0.8; while the AUPRC performance of alternative approaches remains below 0.72 and manifests certain declines as the number of novel non-target anomaly types increases. When the number of new non-target

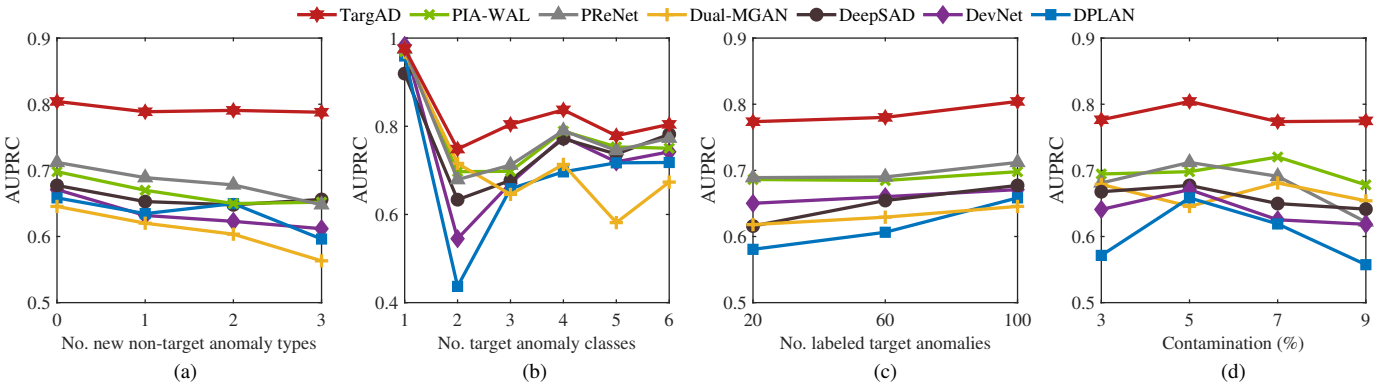


Fig. 4. AUPRC performance of TargAD and the baseline models in identifying target anomalies on the UNSW-NB15 dataset w.r.t. different (a) numbers of new non-target anomaly types (b) numbers of target anomaly classes, (c) numbers of labeled target anomalies and (d) contamination rates.

anomaly types changes from 1 to 2, there is an observed improvement in DPLAN's AUPRC. This improvement can be attributed to the model's ability to leverage deep reinforcement learning, allowing it to extend the learned anomaly patterns. The experimental results in this scenario imply that our model possesses greater adaptability to detect target anomalies in scenarios involving the emergence of novel anomalies.

In practical application scenarios, the number of anomaly classes that are of interest can vary. Therefore, we then proceeded to evaluate the performance of TargAD when confronted with different numbers of target anomaly classes in the training set. Specifically, we systematically varied the number, denoted as m, of target anomaly classes, incrementing it from 1 to 6. Correspondingly, the number of non-target anomaly types was decreased from 6 to 1. Fig. 4(b) illustrates the AUPRC performances of TargAD and other baseline methods for varying numbers of target anomaly classes within the training set. Across all of the above six settings, TargAD consistently outperformed the other baselines in terms of AUPRC. The results confirm that our model has the capability to adapt to diverse detection scenarios irrespective of whether they involve the identification of a single or multiple target anomaly types, and TargAD consistently improves the detection accuracy irrespective of the number of target anomalies. TargAD and the baseline models achieve better results when confronted with a single type of target anomaly compared to situations involving multiple types of target anomalies. The rationale is that a single class of target anomaly provides more advantageous conditions for learning anomaly patterns.

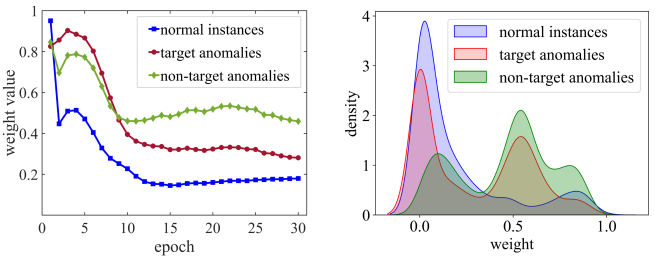
Our model and the semi-supervised baselines utilize a small number of labeled target anomalies to provide prior knowledge concerning anomalies, aimed at imporving detection accuracy. Thus, we next analyze the influence of variations in the number of labeled anomalies within the training set on the model's performance. The labeled anomalies encompass three distinct types (*Generic*, *Backdoor*, and *DoS*), with the quantity of each available labeled anomaly type varying within $\{20, 60, 100\}$. The anomaly contamination rate of the unlabeled data was fixed to be 5%. Fig. 4(c) displays the AUPRC performance of TargAD and the baselines as the number of labeled anomalies changes. TargAD exhibits a certain level of robustness in the face of varying numbers of labeled anomalies while consistently delivering commendable AUPRC performance, even in scenarios where limited quantities of labeled anomalies are available. As the number of labeled anomalies increases, the AUPRC performance of TargAD and the baseline models improves. This is because a larger volume of labeled data generally facilitates to better train the model, thereby endowing the models with more comprehensive insights into each distinct class of target anomalies.

In unlabeled training data, there is often an inherent proportion of anomaly contamination. To assess the robustness of TargAD against varying contamination rates (representing the proportion of anomalies) in unlabeled training data, we conducted experiments using contamination rates selected from the set $\{3\%, 5\%, 7\%, 9\%\}$. Under these four contamination rate settings, we kept the available labeled target anomaly classes to be *Generic*, *Backdoor*, and *Analysis*, with each class comprising 100 instances. The AUPRC results of TargAD and the baselines under different contamination rates are illustrated in Fig. 4(d). Drawing upon the observed results, we can make the following remarks. Despite the presence of different anomaly contamination levels, TargAD consistently outperforms the other baseline models, yielding better AUPRC improvements against each. It is intriguing to observe that both TargAD and the baseline models achieve better target anomaly identification results at mid-range contamination rates (5% or 7%). When the contamination rate of unlabeled data is low, the candidate selection phase of TargAD is adversely affected. This can be attributed to the fixed threshold (α) of 5%, leading to a higher proportion of real normal instances among the selected non-target anomaly candidates. It is easier for the baseline models to learn the characteristics of normal instances under a lower contamination rate, but they lack sufficient focus on anomalous patterns. As the contamination rate continues to rise, TargAD maintains its stability; whereas the baseline models may struggle to cope with the noise due to the weak supervision information of labeled anomalies, resulting in a decline in their AUPRC performance.

*3) Ablation Study (**RQ3**):* To assess the contribution of different loss terms in the $\mathcal{L}_{\text{clf}}$ on the identification of target anomalies, we conducted ablation experiments with

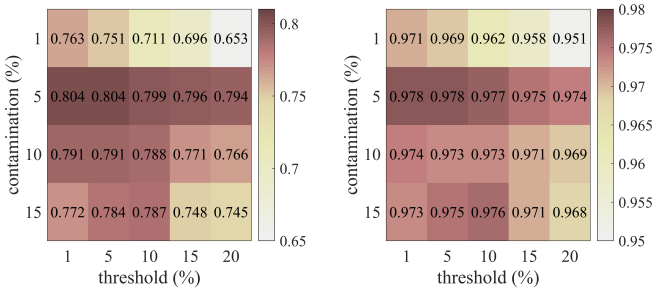
TABLE III
AUROC AND AUPRC PERFORMANCE (± STANDARD DEVIATION) OF TARGAD AND ITS ABLATED VARIANTS ON THE UNSW-NB15 DATASET.

Models	AUPRC	lift(%)	AUROC	lift(%)
$\text{TargAD}_{-\text{O}}$	0.784±0.004	2.0	0.973±0.002	0.5
$\text{TargAD}_{-\text{R}}$	0.786±0.002	1.8	0.966±0.001	1.2
$\text{TargAD}_{-\text{O}-\text{R}}$	0.764±0.003	4.0	0.958±0.003	2.0
TargAD	**0.804±0.001**		**0.978±0.001**	



(a) (b)

Fig. 5. Effects of the designed weight-updating strategy on normal instances, target anomalies, and non-target anomalies. (a) Evolution of weight values for the three types of instances at each epoch. (b) Weight density distributions for the three types of instances at the final epoch.



(a) (b)

Fig. 6. (a) AUPRC and (b) AUROC performance of TargAD on the UNSW-NB15 dataset under different values of the threshold α and the ground truth contamination rates.

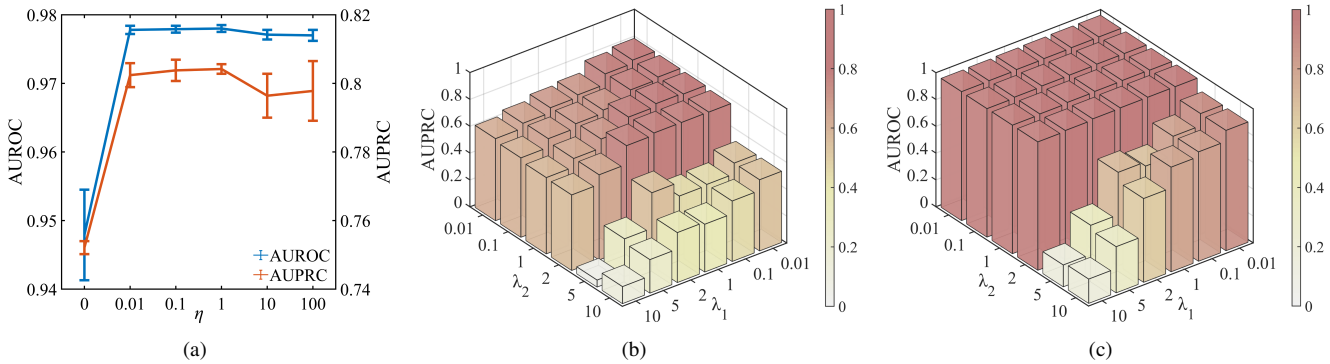
three variants: $\text{TargAD}_{-\text{O}-\text{R}}$, $\text{TargAD}_{-\text{O}}$, and $\text{TargAD}_{-\text{R}}$. $\text{TargAD}_{-\text{O}-\text{R}}$ represents the scenario where the classifier solely uses $\mathcal{L}_{\text{CE}}$ for parameter updates. $\text{TargAD}_{-\text{O}}$ excludes $\mathcal{L}_{\text{OE}}$ from $\mathcal{L}_{\text{clf}}$, while $\text{TargAD}_{-\text{R}}$ excludes $\mathcal{L}_{\text{RE}}$ from $\mathcal{L}_{\text{clf}}$. The results of the three ablated model variants and TargAD on the UNSW-NB15 dataset, in terms of AUROC and AUPRC, are presented in Table III. TargAD exhibits the best performance with respect to both AUROC and AUPRC metrics, showcasing an average improvement ranging from 0.5% to 2% for AUROC, and from 2% to 4% for AUPRC. Notably, $\text{TargAD}_{-\text{O}-\text{R}}$ presents comparatively inferior performance when compared to the exclusion of $\mathcal{L}_{\text{OE}}$ or $\mathcal{L}_{\text{RE}}$ from the loss function of the classifier. This ablation study underscores the significance of the $\mathcal{L}_{\text{OE}}$ and $\mathcal{L}_{\text{RE}}$ terms in the training of the classifier.

*4) Effect of the weight-updating strategy (**RQ4**):* Below we provide a comprehensive analysis of the effectiveness of the weight-updating strategy incorporated in $\mathcal{L}_{\text{OE}}$. Fig. 5(a) presents a visualization depicting the average weights assigned to the three types of instances (inaccurately reconstructed normal instances, target anomalies, and non-target anomalies) for the non-target anomaly candidates at each epoch. The initialization of instance weights occurs during the first iteration of the classifier training, following Eq. (5). Normal instances exhibit comparatively low reconstruction errors, leading to their assignment of higher weights, whereas target anomalies and non-target anomalies are assigned relatively lower weights. Subsequently, weight updates are performed according to Eq. (4), giving higher weights to the non-target anomalies. Fig. 5(a) shows that, by the second epoch, the average weight of the normal instances experiences a significant decrease, ultimately becoming the lowest among the weights of the three instance types. As the training progresses, specifically after the ninth epoch, the average weight of the non-target anomalies surpasses that of the target anomalies and normal instances, and the weights of all three instance types converge to a stable state.

Furthermore, Fig. 5(b) displays a comprehensive illustration of the density distribution of weights for the three instance types during the final epoch. Within the low-weight region, the density peak associated with the non-target anomalies is distinctly lower compared to the other two instance types. In contrast, the distribution of non-target anomalies becomes increasingly concentrated in the high-weight region. The experimental findings unequivocally attest to the effectiveness of the designed weight-updating strategy in assigning higher weights to non-target anomalies. This strategy effectively tackles the issue of noise present among the non-target anomaly candidates.

*5) Parameter sensitivity (**RQ5**):* To investigate TargAD's sensitivity with respect to α, we analyze its robustness under different ground truth contamination rates. Considering different values of α ($\alpha \in \{1\%, 5\%, 10\%, 15\%, 20\%\}$), We run TargAD on the UNSW-NB15 dataset whose unlabeled training data with contamination rates varying from $\{1\%, 5\%, 10\%, 15\%\}$. Fig. 6 presents a matrix showing the average AUPRC and AUROC performance of TargAD as these two variables undergo variations. When the ground truth contamination rate remains constant, TargAD demonstrates robust performance as long as the selected α value remains below the contamination rate. Conversely, surpassing the contamination rate with the chosen α value leads to a consistent decline in TargAD's effectiveness. This decline can be attributed to the erroneous inclusion of more real normal instances as non-target anomaly candidates due to the elevated α value. The predicted probabilities for such normal instances in the classifier training phase converge towards a uniform distribution, consequently hampering the model's overall performance.

We then run experiments on the UNSW-NB15 dataset for $\eta \in \{0, 0.01, 0.1, 1, 10, 100\}$ to analyze the sensitivity of TargAD with respect to the trade-off parameter η. Fig. 7(a) shows the AUROC and AUPRC results of TargAD across various values of the trade-off parameter η. When η is set to 0, indicating that the autoencoders responsible for selecting potential non-target anomaly candidates and normal candidates were not trained using labeled target anomalies, the model's



(a) (b) (c)

Fig. 7. Performance of TargAD with different trade-off parameter values. (a) AUROC (left) and AUPRC (right) performance under varying values of η in $\mathcal{L}_{\mathrm{AE}_i}$. (b) AUPRC performance under different values of λ_1 and λ_2 in $\mathcal{L}_{\mathrm{clf}}$. (c) AUROC performance under different values of λ_1 and λ_2 in $\mathcal{L}_{\mathrm{clf}}$.

TABLE IV
PRECISION, RECALL, AND F1-SCORE PERFORMANCE USING DIFFERENT STRATEGIES IN TARGAD.

	Strategies								
	Maximum Softmax Probability (MSP)			Energy Score (ES)			Energy Discrepancy (ED)		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score
normal instances	0.935	0.972	0.953	0.934	**0.982**	**0.957**	**0.936**	0.970	0.953
target anomalies	0.644	**0.812**	**0.718**	0.571	0.291	0.385	**0.810**	0.438	0.569
non-target anomalies	0.414	0.209	0.278	0.375	0.351	0.362	**0.449**	**0.467**	**0.458**
macro avg	0.665	**0.664**	0.650	0.627	0.541	0.568	**0.732**	0.625	**0.660**
weighted avg	0.861	**0.882**	0.867	0.849	0.866	0.854	**0.877**	0.879	**0.874**

performance significantly deteriorates. As η increases above 0, TargAD exhibits certain robustness to variations in η.

In addition, we conduct experiments while keeping the trade-off parameter $\eta = 1$ and varying the trade-off parameters $\lambda_1, \lambda_2 \in \{0.01, 0.1, 1, 2, 5, 10\}$ to inspect the sensitivity of TargAD to them. The AUPRC and AUROC performances are depicted in Fig. 7(b) and Fig. 7(c), respectively. The results suggest that the loss function for training TargAD classifiers tends to take smaller λ_1 and λ_2 values. When the value of λ_1 or λ_2 exceeds 1, the performance of our model undergoes a decline. Specifically, when λ_1 assumes larger values, the focus of the classifier shifts towards the learning of instances within the non-target anomaly candidates while neglecting adequate attention to instances in $\mathcal{D}_U^N$ and $\mathcal{D}_L$. Therefore, the model's performance suffers a decline. Similarly, a higher value of λ_2 reduces the prediction confidence for non-target anomalies, as they might be present among the normal candidates, thereby leading to a deterioration in the model's performance.

*6) Non-target Anomaly Identification(**RQ6**):* As previously mentioned in Section III-C, TargAD offers a unique capability of distinguishing among normal instances, target anomalies, and non-target anomalies by identifying non-target anomalies as a separate group based on OOD detection strategies. This facilitates flexible adjustment of detection goals to meet specific requirements in practical applications, setting it apart from alternative approaches. We conducted experiments by employing three distinct strategies: MSP [32], ES [37], and ED [43]. The results, presented in Table IV, demonstrate that the ED strategy outperforms MSP and ES in recognizing non-target anomalies with respect to Precision, Recall, and F1-Score metrics. This superiority can be attributed to the fact that ED not only maintains the energy's nature to mitigate the issue of overconfidence but also takes into account the overall distribution of logits. In addition, to assess the overall effectiveness of identifying the three types of instances (normal instances, target anomalies, and non-target anomalies), we applied two averaging methods: macro average and weighted average. The results in Table IV indicate that using the ED strategy yields superior performance over the other two strategies in the comprehensive identification of the three types of instances.

V. CONCLUSION

In this paper, we tackle a crucial yet overlooked practical scenario of anomaly detection in which not all anomalies are of primary interest due to varying risk levels. We introduce TargAD, a semi-supervised anomaly detection model specifically designed to overcome the challenges associated with identifying target anomalies that present severe threats. The model incorporates a selection mechanism to filter out normal and non-target anomaly candidates and utilizes a novel loss function that maximizes the distributional disparities among normal candidates, target anomalies, and non-target anomaly candidates. Extensive experiments demonstrate that the proposed model achieves exceptional performance in effectively identifying target anomalies, significantly outperforming state-of-the-art methods in terms of AUPRC. Compared to the alternatives, TargAD exhibits superior robustness while attaining precise detection even with fewer labeled target anomalies.

ACKNOWLEDGMENT

This work was supported in part by National Key Research and Development Program of China (No. 2021YFB2700100), Shanghai "Science and Technology Innovation Action Plan" Project (No.23511100700), and Program of Shanghai Academic Research Leader (No. 23XD1401100).

REFERENCES

[1] G. Pang, C. Shen, L. Cao, and A. V. D. Hengel, "Deep learning for anomaly detection: A review," *ACM computing surveys (CSUR)*, vol. 54, no. 2, pp. 1–38, 2021.

[2] F. Carcillo, Y.-A. Le Borgne, O. Caelen, Y. Kessaci, F. Oblé, and G. Bontempi, "Combining unsupervised and supervised learning in credit card fraud detection," *Information sciences*, vol. 557, pp. 317–331, 2021.

[3] P.-F. Marteau, "Random partitioning forest for point-wise and collective anomaly detection—application to network intrusion detection," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 2157–2172, 2021.

[4] K. Zhou, Y. Xiao, J. Yang, J. Cheng, W. Liu, W. Luo, Z. Gu, J. Liu, and S. Gao, "Encoding structure-texture relation with p-net for anomaly detection in retinal images," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16*. Springer, 2020, pp. 360–377.

[5] X. Chen, C. Dong, J. Ji, J. Cao, and X. Li, "Image manipulation detection by multi-view multi-scale supervision," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 14 185–14 193.

[6] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen, "Deep autoencoding gaussian mixture model for unsupervised anomaly detection," in *International conference on learning representations*, 2018.

[7] J. Audibert, P. Michiardi, F. Guyard, S. Marti, and M. A. Zuluaga, "Usad: Unsupervised anomaly detection on multivariate time series," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 3395–3404.

[8] X. Jiang, J. Liu, J. Wang, Q. Nie, K. Wu, Y. Liu, C. Wang, and F. Zheng, "Softpatch: Unsupervised anomaly detection with noisy data," *Advances in Neural Information Processing Systems*, vol. 35, pp. 15 433–15 445, 2022.

[9] M. E. Villa-Pérez, M. A. Alvarez-Carmona, O. Loyola-Gonzalez, M. A. Medina-Pérez, J. C. Velazco-Rossell, and K.-K. R. Choo, "Semi-supervised anomaly detection algorithms: A comparative summary and future research directions," *Knowledge-Based Systems*, vol. 218, p. 106878, 2021.

[10] S. Akcay, A. Atapour-Abarghouei, and T. P. Breckon, "Ganomaly: Semi-supervised anomaly detection via adversarial training," in *Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part III 14*. Springer, 2019, pp. 622–637.

[11] L. Ruff, R. A. Vandermeulen, N. Görnitz, A. Binder, E. Müller, K.-R. Müller, and M. Kloft, "Deep semi-supervised anomaly detection," in *International Conference on Learning Representations*, 2020.

[12] G. Pang, L. Cao, L. Chen, and H. Liu, "Learning representations of ultrahigh-dimensional data for random distance-based outlier detection," in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2018, pp. 2041–2050.

[13] G. Pang, C. Shen, and A. van den Hengel, "Deep anomaly detection with deviation networks," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 353–362.

[14] L. Zhang, X. Xie, K. Xiao, W. Bai, K. Liu, and P. Dong, "Manomaly: Mutual adversarial networks for semi-supervised anomaly detection," *Information Sciences*, vol. 611, pp. 65–80, 2022.

[15] C. Douligeris and A. Mitrokotsa, "Ddos attacks and defense mechanisms: classification and state-of-the-art," *Computer networks*, vol. 44, no. 5, pp. 643–666, 2004.

[16] A. O. Ishaya, A. Aminat, B. Hashim, and A. A. Adekunle, "Improved detection of advanced persistent threats using an anomaly detection ensemble approach," *Advances in Science, Technology and Engineering Systems Journal*, vol. 6, no. 2, pp. 295–302, 2021.

[17] X. Hu, T. Wang, M. P. Stoecklin, D. L. Schales, J. Jang, and R. Sailer, "Muse: asset risk scoring in enterprise network with mutually reinforced reputation propagation," *EURASIP Journal on Information Security*, vol. 2014, pp. 1–9, 2014.

[18] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation-based anomaly detection," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 6, no. 1, pp. 1–39, 2012.

[19] Y.-L. Zhang, L. Li, J. Zhou, X. Li, and Z.-H. Zhou, "Anomaly detection with partially observed anomalies," in *Companion Proceedings of the The Web Conference 2018*, 2018, pp. 639–646.

[20] G. Pang, A. van den Hengel, C. Shen, and L. Cao, "Toward deep supervised anomaly detection: Reinforcement learning from partially labeled anomaly data," in *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, 2021, pp. 1298–1308.

[21] D. Hendrycks, M. Mazeika, and T. Dietterich, "Deep anomaly detection with outlier exposure," in *International Conference on Learning Representations*, 2019.

[22] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: identifying density-based local outliers," in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, 2000, pp. 93–104.

[23] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft, "Deep one-class classification," in *International conference on machine learning*. PMLR, 2018, pp. 4393–4402.

[24] Z. Li, Y. Zhao, X. Hu, N. Botta, C. Ionescu, and G. Chen, "Ecod: Unsupervised outlier detection using empirical cumulative distribution functions," *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[25] H. Ju, D. Lee, J. Hwang, J. Namkung, and H. Yu, "Pumad: Pu metric learning for anomaly detection," *Information Sciences*, vol. 523, pp. 167–183, 2020.

[26] Y.-J. Zhang, P. Zhao, L. Ma, and Z.-H. Zhou, "An unbiased risk estimator for learning with augmented classes," *Advances in Neural Information Processing Systems*, vol. 33, pp. 10 247–10 258, 2020.

[27] H. Mu, R. Sun, G. Yuan, and G. Shi, "Positive unlabeled learning-based anomaly detection in videos," *International Journal of Intelligent Systems*, vol. 36, no. 8, pp. 3767–3788, 2021.

[28] B. Tian, Q. Su, and J. Yin, "Anomaly detection by leveraging incomplete anomalous knowledge with anomaly-aware bidirectional gans," in *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, 7 2022, pp. 2255–2261.

[29] W. Zong, F. Zhou, M. Pavlovski, and W. Qian, "Peripheral instance augmentation for end-to-end anomaly detection using weighted adversarial learning," in *Database Systems for Advanced Applications: 27th International Conference, DASFAA 2022, Virtual Event, April 11–14, 2022, Proceedings, Part II*. Springer, 2022, pp. 506–522.

[30] M. Mahdavi, Z. Abedjan, R. Castro Fernandez, S. Madden, M. Ouzzani, M. Stonebraker, and N. Tang, "Raha: A configuration-free error detection system," in *Proceedings of the 2019 International Conference on Management of Data*, 2019, pp. 865–882.

[31] J. Nitsch, M. Itkina, R. Senanayake, J. Nieto, M. Schmidt, R. Siegwart, M. J. Kochenderfer, and C. Cadena, "Out-of-distribution detection for automotive perception," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2021, pp. 2938–2943.

[32] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," in *International Conference on Learning Representations*, 2017.

[33] S. Liang, Y. Li, and R. Srikant, "Enhancing the reliability of out-of-distribution image detection in neural networks," in *International Conference on Learning Representations*, 2018.

[34] Y. Sun, C. Guo, and Y. Li, "React: Out-of-distribution detection with rectified activations," *Advances in Neural Information Processing Systems*, vol. 34, pp. 144–157, 2021.

[35] X. Dong, J. Guo, A. Li, W.-T. Ting, C. Liu, and H. Kung, "Neural mean discrepancy for efficient out-of-distribution detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 19 217–19 227.

[36] H. Wei, R. Xie, H. Cheng, L. Feng, B. An, and Y. Li, "Mitigating neural network overconfidence with logit normalization," in *International Conference on Machine Learning*. PMLR, 2022, pp. 23 631–23 644.

[37] W. Liu, X. Wang, J. Owens, and Y. Li, "Energy-based out-of-distribution detection," *Advances in neural information processing systems*, vol. 33, pp. 21 464–21 475, 2020.

[38] Z. Lin, S. D. Roy, and Y. Li, "Mood: Multi-level out-of-distribution detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 15 313–15 323.

[39] S. Mohseni, M. Pitale, J. Yadawa, and Z. Wang, "Self-supervised learning for generalizable out-of-distribution detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, 2020, pp. 5216–5223.

[40] J. Chen, Y. Li, X. Wu, Y. Liang, and S. Jha, "Atom: Robustifying out-of-distribution detection using outlier mining," in *Machine Learning and Knowledge Discovery in Databases. Research Track: European Conference, ECML PKDD 2021, Bilbao, Spain, September 13–17, 2021, Proceedings, Part III 21*. Springer, 2021, pp. 430–445.

[41] Y. Ming, Y. Fan, and Y. Li, "Poem: Out-of-distribution detection with posterior sampling," in *International Conference on Machine Learning*. PMLR, 2022, pp. 15 650–15 665.

[42] Q. Wang, J. Ye, F. Liu, Q. Dai, M. Kalander, T. Liu, J. Hao, and B. Han, "Out-of-distribution detection with implicit outlier transformation," in *International Conference on Learning Representations*, 2023.

[43] R. He, Z. Han, X. Lu, and Y. Yin, "Safe-student for safe deep semi-supervised learning with unseen-class unlabeled data," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 14 585–14 594.

[44] Y. Zhou, X. Song, Y. Zhang, F. Liu, C. Zhu, and L. Liu, "Feature encoding with autoencoders for weakly supervised anomaly detection," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 6, pp. 2454–2465, 2021.

[45] Z. Li, C. Sun, C. Liu, X. Chen, M. Wang, and Y. Liu, "Dual-mgan: An efficient approach for semi-supervised outlier detection with few identified anomalies," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 16, no. 6, pp. 1–30, 2022.

[46] G. Pang, C. Shen, H. Jin, and A. van den Hengel, "Deep weakly-supervised anomaly detection," in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023, pp. 1795–1807.