Peripheral Instance Augmentation for End-to-End Anomaly Detection using Weighted Adversarial Learning

Weixian Zong[1], Fang Zhou[1⋆], Martin Pavlovski[2], and Weining Qian[1]

[1] School of Data Science and Engineering,
East China Normal University, Shanghai, China
wxzong@stu.ecnu.edu.cn, {fzhou,wnqian}@dase.ecnu.edu.cn
[2] Temple University, Philadelphia, USA
martin.pavlovski@temple.edu

Abstract. Anomaly detection has been a lasting yet active research area for decades. However, the existing methods are generally biased towards capturing the regularities of high-density normal instances with insufficient learning of peripheral instances. This may cause a failure in finding a representative description of the normal class, leading to high false positives. Thus, we introduce a novel anomaly detection model that utilizes a small number of labeled anomalies to guide the adversarial training. In particular, a weighted generative model is applied to generate peripheral normal instances as supplements to better learn the characteristics of the normal class, while reducing false positives. Additionally, with the help of generated peripheral instances and labeled anomalies, an anomaly score learner simultaneously learns (1) latent representations of instances and (2) anomaly scores, in an end-to-end manner. The experimental results show that our model outperforms the state-of-the-art anomaly detection methods on four publicly available datasets, achieving improvements of 6.15%-44.35% in AUPRC and 2.27%-22.3% in AUROC, on average. Furthermore, we applied the proposed model to a real merchant fraud detection application, which further demonstrates its effectiveness in a real-world setting.

1 Introduction

Anomaly detection, referred to as the process of identifying unexpected patterns from the normal behavior in a dataset, can provide critical help in various applications, such as fraud detection in finance [9], network attack detection in cybersecurity [36] and disease detection in healthcare [27]. A plethora of anomaly detection methods has been introduced over the years. However, such methods mainly focus on improving the accuracy of detecting anomalies with insufficient attention to high false positives. In real applications, when a great number of normal instances are incorrectly reported as anomalies, this may lead to a waste of labor and material resources.

[⋆] Corresponding author

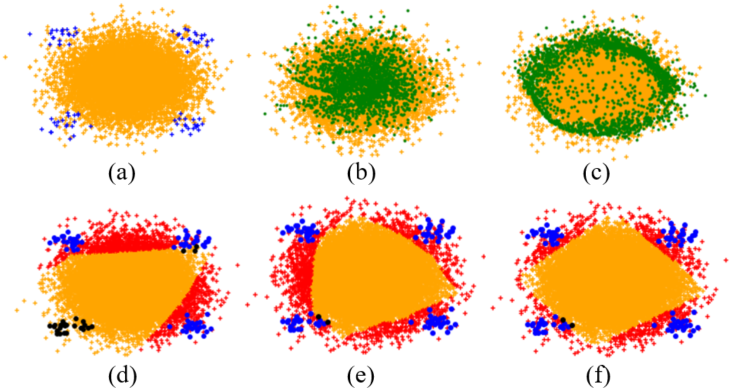

Fig. 1. (a) Synthetic data composed of normal data (colored in orange) as well as anomalies (colored in blue). (b) and (c) show the instances (colored in green) generated by a conventional generative model (WGAN-GP) and our model, respectively. (d), (e) and (f) show the performance of anomaly detection by two recent anomaly detection models, GANomaly and DevNet, and our model PIA-WAL. The false positives and false negatives are colored in red and black, respectively.

Let us take a merchant fraud detection problem as an example. When certain merchants are considered to be involved in fraudulent activities, a subsequent on-the-spot investigation needs to be conducted to verify whether the merchants were indeed involved in fraudulent events. Suppose that a model outputs a huge amount of potential fraudulent merchants (e.g., $> 15,000$ normal merchants were predicted as anomalies (see Table 3)), enormous resources are required for investigation and verification, in order to maintain a fair and secure financial environment. Consequently, reducing the number of false positives while enhancing anomaly detection accuracy is a rather challenging task.

In the last decade, research efforts have been focused on unsupervised anomaly detection methods [15, 19]. However, such methods typically produce a large amount of false positives due to lack of labeled data. Considering that sufficient amounts of labeled normal instances can be easily collected, many deep learning approaches were proposed to capture the regularities of normal instances [6]. For instance, GAN-based methods [31, 23] aim to learn a latent feature space to capture the normality underlying the given data. On the other hand, a few labeled anomalous data with valuable prior knowledge are available in many real-world applications. Approaches such as DevNet [22] utilize labeled anomalies to guarantee a margin between anomalies and normal instances so as to improve detection accuracy. However, these methods may learn suboptimal representations of normal instances as they fail to capture the characteristics of peripheral normal instances, thus leading to high false positives.

To illustrate the aforementioned issue, we present an example in Fig. 1. Fig. 1(a) displays a 2-dimensional synthetic dataset, where the orange dots represent the normal instances and the blue dots represent anomalies. The green dots in Fig.1(b) represent the instances generated by a conventional generative model (WGAN-GP [13]). Evidently, most generated instances are mainly concentrated in the center, which implies that the generative model is biased towards learning features of high-density instances, while overlooking peripheral instances that account for a small portion of the normal class. A mass of methods [20, 22, 24] suffer from the same problem, which leads to a high number of false positives. An illustrated explanation is presented in Fig. 1(d) and 1(e), which show the prediction results of two recent anomaly detection methods, GANomaly [1] and DevNet [22], respectively. It is obvious that a large amount of peripheral normal instances were incorrectly reported as anomalies (colored in red) by both methods. In addition, compared to Fig. 1(e), more anomalies were incorrectly predicted as normal instances (colored in black) in Fig. 1(d) as GANomaly does not take labeled anomalies into account.

In this work, we develop a weighted generative model by leveraging a few labeled anomalies for anomaly detection, named PIA-WAL (Peripheral Instance Augmentation with Weighted Adversarial Learning). The goal of our model is to learn representative descriptions of normal instances in order to reduce the amount of false positives while maintaining accurate anomaly detection. The model consists of two modules, an anomaly score learner and a weighted generative model. The anomaly score learner utilizes a small number of labeled anomalies to distinguish anomalies from normal instances. The scores outputted by the anomaly score learner are expected to reflect the difficulty of instances to be correctly classified. Since the peripheral normal instances constitute a small portion of the data, they are generally challenging to capture and are hard to be correctly predicted. To address this issue, we introduce a weighted generative model guided by the outputs of the anomaly score learner to generate additional *peripheral normal instances* as supplements (see the green dots in Fig.1(c)), in order to assist the anomaly score learner to better find a representative description of the normal class. Fig. 1(f) clearly shows that our model is able to produce a high detection accuracy as well as low false positives.

The advantages of the proposed model are threefold. First, by generating peripheral normal instances as supplements, PIA-WAL is able to capture the complex feature space of normal instances while reducing false positives. Second, PIA-WAL couples the feature representation learning and the anomaly score optimization in an end-to-end schema. Third, PIA-WAL is consistently more robust than the other anomaly detection methods under various anomaly contamination levels.

To verify the effectiveness and robustness of PIA-WAL, we conducted experiments on four publicly available datasets. The results show that PIA-WAL outperforms seven state-of-the-art methods, achieving statistically significant improvements of 2.27%-22.3% in AUROC and 6.15%-44.35% in AUPRC, on average. Furthermore, we applied PIA-WAL to real-world merchant fraud detection, the results of which further demonstrate its effectiveness.

This work makes the following major contributions:
(1) To the best of our knowledge, this work is the first to integrate a small amount of labeled anomalies into an adversarial framework to achieve end-to-end anomaly score learning.
(2) A novel anomaly detection model, namely PIA-WAL, is introduced. With the help of generating peripheral normal instances, the detector can find a more representative description of the normal class while accurately detecting anomalies. The code of PIA-WAL will be available online (not posted yet to facilitate double-blind review).
(3) The results obtained on publicly available datasets and a real-world merchant fraud dataset demonstrate that PIA-WAL achieves substantial improvements over state-of-the-art methods.

2 Related Work

Anomaly Detection Anomaly detection is the task of identifying anomalies that deviate significantly from the majority of data objects [21]. Most conventional approaches are unsupervised including distance-based [4], density-based [7], isolation-based methods [15], and so forth. However, such methods are often ineffective in handling high-dimensional or irrelevant features. Recently, deep learning has been explored to improve anomaly detection [3, 5]. Autoencoder-based methods [8, 35] utilize a bottleneck network to learn a low-dimensional representation space and then use the learned representations to define reconstruction errors as anomaly scores. Anomaly measure-dependent learning methods [19, 29] aim at learning feature representations that are specifically optimized for a particular anomaly measure. These deep learning methods can capture more complex feature interactions compared to the traditional shallow methods, however, they learn the feature representations separately from the subsequent anomaly detection, leading to suboptimal detection performance.

The recent advances show that deep anomaly detection can be substantially improved when labeled anomalies are utilized to guarantee a margin between the labeled anomalies and normal instances [25]. For example, ADOA [33] tries to cluster the observed anomalies into k clusters, and detect potential anomalies and reliable normal instances from unlabeled instances. DevNet [22] leverages a few labeled anomalies and a Gaussian prior over anomaly scores by defining a deviation loss to perform end-to-end anomaly detection. Deep SAD [24] enforces a margin between the one-class center and the labeled anomalies while minimizing the center-oriented hypersphere. With the help of prior knowledge of anomalies, these models can achieve better detection performance. However, due to the peripheral normal instances constituting a small portion of the training set, these models may fail to capture the characteristics of such instances. This makes it difficult to find a representative description of the normal class, leading to a large number of false positives.

GAN-based Anomaly Detection Generative Adversarial Networks (GANs) [12] and the adversarial training framework have been successfully applied to model complex and high-dimensional distributions of real-world data. This GANs' characteristic suggests they can be used for anomaly detection, although their applications have been only recently explored [10, 23]. Anomaly detection using GANs is the task of modeling the normal behavior using the adversarial training process and detecting anomalies by defining some form of a residual between a real instance and a generated instance as an anomaly score. One of the earliest methods is AnoGAN [26] which involves training a standard GAN on normal instances only to enforce the generator to learn the manifold of normal instances. Since the latent space can capture the underlying distribution of normal data, anomalies are expected to be less likely to have highly similar generated counterparts compared to normal instances. One major issue with AnoGAN is the computational inefficiency in the iterative search for the most similar generated instance to the input instances. The model EGBAD [31] and its variant [32] were designed to address this problem by learning the inverse mapping of raw data from the latent space. GANomaly [1] further improves the generator over the previous work by replacing the generator network with an encoder-decoder-encoder network and adding two additional loss functions. Different from the aforementioned methods, OCAN [34] and Fence GAN [18] generate fringe instances that are complementary, rather than matching, to the normal data which are used as reference anomalies. However, it cannot guarantee that the generated reference instances well resemble the unknown anomalies. Unlike OCAN, our model generates instances that are close to hard-to-classify normal instances and thus act as normal class supplements.

3 Preliminary: Wasserstein GAN with Gradient Penalty (WGAN-GP)

Generative adversarial networks (GANs) constitute a powerful class of generative models involving an adversarial process in which two modules, a generator G and a discriminator D, are trained simultaneously. The generator G models high dimensional data from a prior noise distribution P_z to learn the real data distribution P_r, while the discriminator D is a binary classifier that estimates the probability that a instance is from the real data x rather than the generated fake data $G(z)$. The objective function of a GAN is : $\min_{G} \max_{D} \mathbb{E}_{x \sim P_r} [logD(x)] + \mathbb{E}_{z \sim P_z} [log(1 - D(G(z)))]$.

However, the Jensen-Shannon divergence that GANs aim to minimize is potentially not continuous with respect to the generator's parameters, leading to training difficulties. To solve this problem, [2] propose using the Earth-Mover distance instead in the loss function of the original GAN, thus resulting in the so-called Wasserstein GAN (WGAN) that improves training stability. Under mild assumptions, the Earth-Mover distance is continuous everywhere and differentiable almost everywhere. Thus, WGAN's objective function is constructed using the Kantorovich-Rubinstein duality[28]:

$$\min_{G} \max_{D \in \mathcal{D}} \mathop{\mathbb{E}}_{x \sim P_r} [D(x)] - \mathop{\mathbb{E}}_{z \sim P_z} [D(G(z))] , \tag{1}$$

where $\mathcal{D}$ is the set of 1-Lipschitz functions and minimizing Eq. (1) with respect to the generator parameters essentially minimizes the Earth-Mover distance between the two distributions. To enforce the Lipschitz constraint on the discriminator, WGAN simply clips the weights of the critic (that is, a discriminator variant) to lie within a compact space $[-c, c]$.

However, WGAN's weight clipping always results in either vanishing or exploding gradients without careful tuning of the clipping threshold c. Considering these issues, WGAN-GP [13] is proposed by introducing gradient penalty to enforce the Lipschitz constraint. The objective function is defined with an additional term forcing the gradient to be smaller than 1:

$$L^{Disc} = \mathop{\mathbb{E}}_{z \sim P_z} [D(G(z))] - \mathop{\mathbb{E}}_{x \sim P_r} [D(x)] + \lambda \mathop{\mathbb{E}}_{\hat{x} \sim P_{\hat{x}}} \left[(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2 \right] \tag{2}$$

$$L^{Gen} = - \mathop{\mathbb{E}}_{z \sim P_z} [D(G(z))] , \tag{3}$$

where λ is the penalty coefficient and $P_{\hat{x}}$ is a uniform sampling distribution along straight lines between pairs of points sampled from the original data distribution and the generated data distribution.

4 Method

4.1 Problem Statement

Assume a dataset of m training instances $X = \{x_1, x_2, ..., x_l, x_{l+1}, ..., x_m\}$ with $x_i \in R^D$, where $X^A = \{x_1, x_2, ..., x_l\}$ is a very small set of labeled anomalies and $X^N = \{x_{l+1}, x_{l+2}, ..., x_m\}$ is a set of normal instances, $|X^A| \ll |X^N|$. The goal is to identify whether an instance is anomalous or not.

4.2 Proposed Framework

In this section, we present the details of the proposed model, PIA-WAL. As shown in Fig. 2, the model consists of two major modules, *an anomaly score learner* and *a weighted generative model*. The anomaly score learner, trained on real (and generated) normal instances and a small amount of labeled anomalies, aims to distinguish anomalies from normal instances. A weighted generative model is introduced to assist the anomaly score learner by generating supplemental instances that are close to the peripheral normal instances. With the inclusion of the generated instances, the anomaly score learner is capable to better differentiate between peripheral normal instances and anomalies.

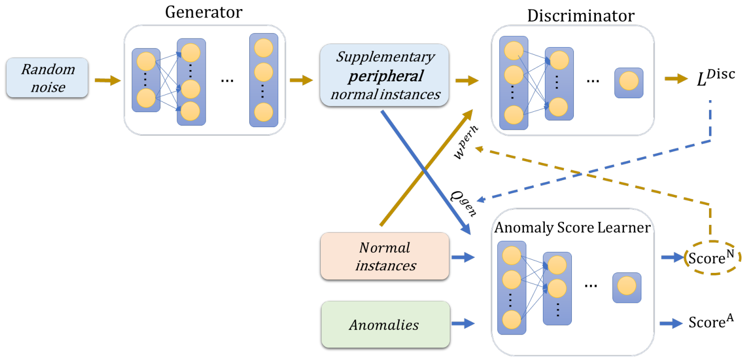

Fig. 2. The framework of PIA-WAL.

Anomaly Score Learner Inspired by the DevNet model [22] suggesting that a small number of labeled anomalies can be leveraged into the mechanism of anomaly scoring, the anomaly score learner in our model draws on a similar idea. Specifically, the learner yields an anomaly score for every given input, and defines a reference score for guiding the subsequent anomaly score learning. The scores of anomalies are enforced to significantly deviate from those of normal instances and lie in the upper tail of the score distribution. Considering that the Gaussian distribution fits the anomaly scores quite well in a variety of datasets [14], we simply use the standard Gaussian distribution as a prior $S \sim N(0,1)$ and set the reference score to 0. The loss function of the anomaly score learner is designed as follows:

$$\ell = \mathop{\mathbb{E}}_{x \in X^N} \left[|\varphi(x)| \right] + \mathop{\mathbb{E}}_{x \in X^A} \left[max(0, \alpha - \varphi(x)) \right], \tag{4}$$

where $\varphi(\cdot)$ is the output of the anomaly scoring network. The loss in Eq. (4) aims to push the scores of normal instances as close as possible to 0, while enforcing the scores of anomalies to deviate from 0 (that is the mean of the normal instances' scores) at least by α. Note that if an anomaly has a negative anomaly score, the loss will be particularly large. Minimizing the loss encourages large positive deviations for all anomalies. Therefore, the learner can enforce the scores of anomalies to significantly deviate from those of normal instances and to lie in the upper tail of the score distribution. To some degree, the scores of normal instances implicitly indicate the difficulty of the instances to be correctly classified. A normal instance with a high anomaly score indicates that it may be a peripheral normal instance that is hard to be detected by the learner. In addition, there is another reasonable possibility that it may be a noisy instance. Thus, we utilize the anomaly scores for guiding the learning process of the subsequent generative model.

Weighted Generative Model As previously discussed, most anomaly detection methods do not focus on learning peripheral normal instances due to their complexity and small proportion, which leads to an increase in false positives.

Thus, we propose a weighted generative model to generate instances that are close to peripheral normal instances, and thus serve as normal class supplements, to improve the anomaly detection accuracy and reduce false positives. In order to achieve guided instance generation, we utilize the outputs of the anomaly score learner to calculate the degree to which a normal instance is *peripheral*, denoted by w^{perh}. The weight w^{perh} of an observed normal instance $x \in X^N$ is defined as:

$$w^{perh}(x) = |\varphi(x)| * \mathbb{I}[|\varphi(x)| < \alpha], \tag{5}$$

where $\mathbb{I}$ is an indicator function and $|\varphi(x)|$ is the absolute value of an observed normal instance's anomaly score. The more difficult it is for the anomaly score learner to distinguish a normal instance x from the anomalies, the higher the value of $\varphi(x)$ is. Considering that the normal class may contain some anomaly contamination in real-world applications, we use an indicator function to guide the model to not be affected by noise during the learning process. When $|\varphi(x)|$ of an observed normal instance exceeds α (the threshold, same used in Eq. (4)), it indicates that the instance is probably a noisy instance in the proximity of anomalies. The value of $w^{perh}(x)$ is then set to 0, thus not affecting the learning of the generative model.

As mentioned in Section 3, the objective of conventional generative models is to minimize the distance between the fake and real data distributions. To make the peripheral normal instances play a more important role in guiding the generator learning, we incorporate the weight w^{perh} into the discriminator loss. We utilize the WGAN-GP model as our basic generative model, due to its excellent data generation capability and stable training. The discriminator loss of WGAN-GP, weighted by w^{perh}, is:

$$\begin{aligned} L^{Disc} &= \mathop{\mathbb{E}}_{z \sim P_z, x \sim P_r} \left[w^{perh}(x) * (D(G(z)) - D(x)) \right] + \\ &\qquad \lambda \mathop{\mathbb{E}}_{\hat{x} \sim P_{\hat{x}}} \left[w^{perh}(x) * (\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2 \right] \\ &= \mathop{\mathbb{E}}_{z \sim P_z} \left[w^{perh}(x) * D(G(z)) \right] - \mathop{\mathbb{E}}_{x \sim P_r} \left[w^{perh}(x) * D(x) \right] \\ &\qquad + \lambda \mathop{\mathbb{E}}_{\hat{x} \sim P_{\hat{x}}} \left[w^{perh}(x) * (\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2 \right]. \end{aligned} \tag{6}$$

Balanced Sampling The real peripheral normal instances may account for an extremely small portion of a mini-batch, thus the role of w^{perh} will be weakened in Eq. (6). To this end, we ensure that the distribution of w^{perh} within a batch is uniform by employing a sampling procedure. We first calculate the w^{perh} of a real normal instance x and scale it within the $[0, 1]$ range by $tanh(\cdot)$ for convenient comparison. We then sample ξ uniformly from $[0, 1]$. If $tanh(w^{perh}(x))$ is higher than ξ, the normal instance x is selected and included into the mini-batch. This step is repeated until the mini-batch is full. The instance selection for the next batch continues from the position of the last selected normal instance. Since ξ is uniformly sampled from $[0, 1]$, the distribution of the w^{perh} weights of the selected normal instances is correspondingly uniform. The normal instances with high w^{perh} weights can indeed play an important role in the generator learning.

Generator's Quality The generated instances are used as supplementary training instances to assist the anomaly score learner to better capture the feature information of the peripheral instances. However, the generated low-quality instances may mislead the anomaly score learner. Therefore, when updating the anomaly score learner's parameters using the generated instances, the quality of the generated instances should be taken into account. In fact, the generated instance's quality is closely related to the generator's quality. Fortunately, because the WGAN model updates the discriminator multiple times before each generator update, it has been shown that the loss function at this point correlates well with the instances' quality [2]. A lower discriminator loss value implies that a generator is capable of generating instances of better quality. The generator's quality is computed as:

$$Q^{gen} = e^{-|L^{Disc}|}. \tag{7}$$

The weights of the generated instances are set to Q^{gen}. With the inclusion of generated instances as normal class supplements in the training set, the learner's loss function is modified as follows:

$$L^{score} = \mathop{\mathbb{E}}_{x \in X^N} [|\varphi(x)|] + \mathop{\mathbb{E}}_{x \in X^A} [max(0, \alpha - \varphi(x))] + Q^{gen} * \mathop{\mathbb{E}}_{x \in G(z)} [|\varphi(x)|] \, . \tag{8}$$

4.3 Outline of PIA-WAL

In summary, the overall objective function of PIA-WAL can be written as follows:

$$\begin{aligned} &\text{PIA-WAL's anomaly score learner}: \quad \min \; L^{score}. \\ &\text{PIA-WAL's weighted generative model}: \min_G \; L^{Gen}, \quad \max_D \; -L^{Disc}. \end{aligned} \tag{9}$$

The training procedure of PIA-WAL is outlined in Algorithm 1. Given a training dataset, in the first iteration, mini-batches are used to update the parameters of the anomaly score learner so as to minimize ℓ in Eq. (4) (Lines 4-8). Note that each mini-batch is constructed by sampling the same number of abnormal and normal instances. Then in the subsequent iterations, the weighted generative model (Lines 10-14) and anomaly score learner (Lines 15-20) are trained in turns. The outputs of the anomaly score learner trained in the previous iteration are used to calculate the degree to which the instances are peripheral (Line 12), thus assisting the weighted generative model learning. The instances generated by the latest updated generative model act as supplementary training instances to further optimize the learner with the corresponding generator's quality (Line 15). By jointly updating the two modules alternately, a trained anomaly score learner is obtained.

During the testing stage, the optimized anomaly score learner is used to produce an anomaly score for every test instance. Here, a reasonable threshold should be selected to determine whether an instance is normal or not. Due to the standard Gaussian prior assumption of the anomaly scores, we use the scores' quantiles to determine a threshold with a desired confidence level[22]. By

Algorithm 1 PIA-WAL's Training Procedure

Input: Dataset $X = X^A \cup X^N$, margin parameter α, penalty coefficient λ, batch size t, number of training epochs for anomaly score learner $Epoch_{learner}$ and generative model $Epoch_{gen}$

Output: A trained anomaly score learner and a weighted generative model

```
Randomly initialize the parameters of the learner and weighted generative model
$n\_batches = (int)(m/t)$
for i=1 to $Epoch_{learner}$ do
    if i==1 then
        for j=1 to n_batches do
            Construct a mini-batch by randomly sampling $t$ instances from $X^A$ and $X^N$, separately
            Update the parameters of the anomaly score learner using Eq. (4)
        end for
    else
        for j=1 to $Epoch_{gen}$ do
            Sample a balanced mini-batch from $X^N$
            Compute $w^{perh}$ for each instance in mini-batch using Eq. (5)
            Optimize the discriminator $D$ and generator $G$ using Eq. (6) and Eq. (3), respectively
        end for
        Calculate generator's quality $Q^{gen}$ using Eq. (7)
        for j=1 to n_batches do
            Use the latest updated generator $G$ to generate $t$ supplementary instances
            Construct a batch with $t$ generated instances, $t$ observed normal instances and $t$ anomalies
            Optimize the anomaly score learner using Eq. (8)
        end for
    end if
end for
```

applying a confidence level of 99.9%, an instance with an anomaly score of over 3.09 ($z_{0.999} = 3.09$) is considered an anomaly, meaning that the probability of an instance being normal is 0.001. Thus, we can set an appropriate and interpretable threshold to identify anomalies with a high confidence level.

5 Experiments

5.1 Datasets and Baselines

In order to verify the effectiveness of PIA-WAL, we conducted experiments on four widely-used real-word datasets. The NB-15 dataset [17] is a network intrusion dataset containing 107,687 data instances, each being 196-dimensional, in which various types of network attacks are treated as anomalies (21.5%). The Census data [11] extracted from the US Census Bureau database contains

299,285 instances in a 500-dimensional space. The task is to detect the rare high-income individuals, which constitute about 6% of the data. The Celeba dataset [16] contains 201,690 celebrity images in a 39-dimensional space in which the scarce bald celebrities (less than 3%) are treated as anomalies. The Fraud data [9] contains 284,807 credit card transactions in a 29-dimensional space. The task is to detect fraudulent transactions, accounting for 0.17%. We split each dataset into training and test sets with a ratio of 8:2. Since only a small number of labeled anomalies are available in many applications, the number of labeled anomalies used for training was fixed to 70 across all datasets in accordance with the setting in [20].

PIA-WAL is compared against 7 state-of-the-art methods. The first three are GAN-based methods, the fourth is a classical unsupervised approach, while the other three leverage labeled anomalies:

- **ALAD** [32] builds upon bi-directional GANs and uses reconstruction errors to determine if instances are anomalous.
- **GANomaly** [1] employs an adversarial autoencoder within an encoder-decoder-encoder pipeline to capture the distribution of training instances in the original and latent spaces.
- **OCAN** [34] leverages the idea of complementary GANs to generate fringe instances used as reference anomalies and train a one-class discriminator when only normal instances are observed.
- **iForest** [15] is a widely-used unsupervised method that detects anomalies based on the number of steps required to isolate instances by isolation trees.
- **ADOA** [33] follows a two-stage procedure. First, the observed anomalies are clustered while the unlabeled instances are categorized into potential anomalies and reliable normal instances. Then, a weighted classifier is trained for further detection.
- **DevNet** [22] is a deep supervised method that leverages a few labeled anomalies with a Gaussian prior to perform an end-to-end differentiable learning of anomaly scores.
- **Deep SAD** [24] minimizes the distance of normal instances to the one-class center while maximizing the distance of known anomalies to the center.

Experimental Setting The hyperparameters of the baselines were set to the same values as the ones used in their respective papers. The anomaly score learner used in DevNet was also leveraged in our model and run with one hidden layer of 20 units and a ReLU activation. We optimized the model parameters using RMSprop with a learning rate of 0.0001, batch size $t = 128$, $Epoch^{learner} = 30$ and $Epoch^{gen} = 1000$. We set the margin $\alpha = 5$ to achieve a very high significance level for all labeled anomalies. The penalty coefficient λ was set to 10 and PIA-WAL's parameters were initialized randomly. All models were implemented in Keras 2.2.4 and ran on a machine with 32GB of memory, 6 CPU cores and 1 quadro p400 GPU.

Two widely-used complementary performance metrics, Area Under the Receiver Operating Characteristic Curve (AUROC) and Area Under the Precision-

Table 1. AUROC and AUPRC performance (with ± standard deviation) of PIA-WAL and the baselines.

Models	AUPRC						AUROC					
	NB-15	Census	Celeba	Fraud	Average	P-value	NB-15	Census	Celeba	Fraud	Average	P-value
ALAD	70.3±4.0	10.3±1.0	6.8±0.8	44.1±2.7	32.88	<0.0001	87.5±0.7	70.1±2.8	76.9±3.6	95.9±0.5	82.60	<0.0001
GANomaly	73.7±1.6	10.3±2.0	8.3±2.4	42.7±7.8	33.75	<0.0001	86.2±0.9	72.2±3.2	79.0±5.6	94.4±2.1	82.95	<0.0001
OCAN	43.3±3.8	8.3±5.1	4.0±1.2	39.4±8.8	23.75	<0.0001	77.5±1.8	62.3±3.0	62.3±8.2	94.7±1.0	74.20	<0.0001
IForest	24.4±1.7	7.2±0.3	6.4±1.6	16.9±4.8	13.73	<0.0001	57.8±2.3	60.1±2.1	70.7±0.9	95.4±0.3	71.00	<0.0001
ADOA	40.2±3.1	23.9±3.5	29.8±1.0	38.0±0.5	32.98	<0.0001	77.8±2.7	84.5±1.4	90.4±0.4	95.9±0.2	87.15	<0.0001
DevNet	81.0±1.6	36.4±0.9	23.6±1.4	62.8±1.0	50.95	0.0001	92.6±0.3	81.2±2.8	94.5±0.2	95.8±1.3	91.03	<0.0001
Deep SAD	**88.8±0.2**	24.2±6.4	22.6±2.4	**72.1±1.5**	51.93	0.006	**95.7±0.1**	73.2±9.2	92.5±1.4	95.4±1.3	89.20	0.0001
PIA-WAL	**88.8±0.7**	**41.5±1.2**	**30.9±1.6**	71.1±0.6	**58.08**	-	95.1±0.2	**85.3±1.1**	**95.5±0.8**	**97.3±0.7**	**93.30**	-

Recall Curve (AUPRC), were used. The paired Wilcoxon signed rank test [30] was used to examine the significance of the performance of PIA-WAL against the seven baselines. All reported results were averaged over 10 independent runs.

5.2 Results and Discussion

Effectiveness on Real-world Datasets Table 1 shows the performances of PIA-WAL and all baselines on the four datasets in terms of AUROC and AUPRC. It is evident that PIA-WAL obtains substantial improvements over the alternative methods. Particularly, across all datasets, PIA-WAl obtains 27.95% higher average AUPRC than GAN-based methods (ALAD, GANomaly and OCAN), 44.35% higher average AUPRC than the unsupervised method (iForest) and 12.79% higher average AUPRC than supervised methods (ADOA, DevNet and Deep SAD). In terms of AUROC, PIA-WAL produces 13.38% and 22.3% higher average AUROC compared to classic GAN-based methods and iForest, respectively. In addition, PIA-WAL also outperforms the supervised methods, namely, DevNet by 2.27% and ADOA by 6.15%. Compared with Deep SAD, PIA-WAL obtains comparable AUPRC on the NB-15 dataset, yet achieves obvious improvements on Census and Celeba and produces a higher AUROC on the Fraud dataset. The improvements are all statistically significant at a 99% confidence level. The reason lies in PIA-WAL's ability to efficiently leverage a limited amount of available anomalies as prior knowledge to enhance the ability of distinguishing anomalies from normal instances. Meanwhile, with the help of generated peripheral normal instances, the model can further reduce the number of false positives.

Robustness under Anomaly Contamination To investigate the robustness of PIA-WAL, we polluted the normal training instances from the NB-15 dataset with contamination levels ranging from 0% up to 20%. The results in terms of AUPRC and AUROC are presented in Fig. 3. The detection performance of all methods decreases with increased contamination levels, particularly for the methods trained using only normal instances (e.g., GANomaly and ALAD). Nevertheless, it is obvious that PIA-WAL consistently outperforms its alternatives across different contamination levels. Moreover, as the contamination level increases, the decline rate of PIA-WAL's performance is much lower than that of Deep SAD; even though they perform comparably when the data is pure. This

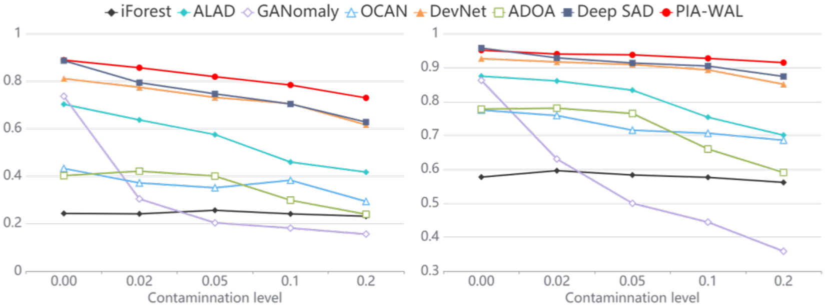

Fig. 3. AUPRC (left) and AUROC (right) under different contamination levels on the NB-15 dataset.

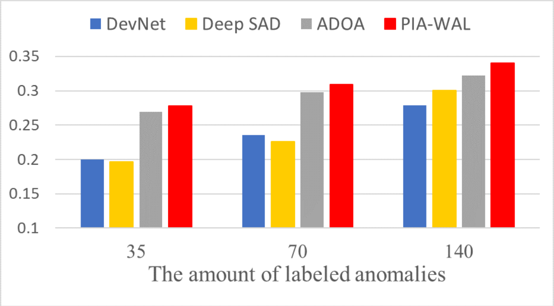

Fig. 4. AUPRC w.r.t the amount of labeled anomalies on the Celeba dataset.

suggests that PIA-WAL has a substantial capability of distinguishing anomalies from normal instances in challenging noisy environments.

Effectiveness w.r.t. the Amount of Labeled Anomalies To account for the difficulty of obtaining labeled anomalies in most applications, we vary the number of labeled anomalies from 35 to 140 and examine the effectiveness of PIA-WAL. Considering that (1) classical GAN-based anomaly detection methods are trained only on normal instances and (2) iForest is an unsupervised method, their performance is invariant to the amount of labeled anomalies. Fig. 4 shows the AUPRCs obtained by the other four methods w.r.t. different numbers of available labeled anomalies on the Celeba dataset. The performance of these methods generally increases with the increased number of labeled anomalies. As the amount of labeled anomalies increases, it is obvious that PIA-WAL consistently outperforms the alternative models. It should be emphasized that, even when using only 35 labeled anomalies, PIA-WAL still achieves the best performance.

Ablation Study To study the impact of Q^{gen} (the weights of the generated instances in Eq. (8)) and w^{perh} (the weights of normal instances in Eq. (6)), we further conducted an ablation study of PIA-WAL on the NB-15 dataset, and reported the results in Table 2. When we disable Q^{gen} in PIA-WAL, there is a drop in performance, as expected, since the anomaly score learner is most probably misled by the generated low-quality instances. When we remove w^{perh}, the generator works similarly to a conventional generative model, rather than

Table 2. Model ablation study results on the NB-15 dataset.

Model	AUPRC	AUROC
PIA-WAL	88.8	95.1
PIA-WAL$_{-Q^{gen}}$	87.7	94.4
PIA-WAL$_{-w^{perh}}$	81.1	94.1

Table 3. Merchant fraud detection performance obtained by the baselines and PIA-WAL.

Model	TN	FP	FN	TP	AUPRC
OCAN	127,151±13,249	33,544±13,249	6±5	100±5	1.2±0.3
ALAD	128,639±1	32,056±1	1±0	**105±0**	4.7±0.8
GANomaly	146,171±3,351	14,524±3,351	9±2	97±2	1.1±0.1
ADOA	153,913±2,051	6,782±2,051	17±11	89±11	2.7±1.4
IForest	154,279±343	6,416±343	23±2	83±2	2.0±0.1
DevNet	159,774±152	921±152	18±3	88±3	41.6±2.3
Deep SAD	-	-	-	-	54.3±3.2
PIA-WAL	160,215±123	**480±116**	20±3	86±3	**56.6±0.8**

generating mainly peripheral instances. The performance decreases by 7.7% in AUPRC. This suggests that the use of generated peripheral normal instances can effectively improve the detection accuracy.

6 Application to Merchant Fraud Detection

We applied PIA-WAL to real-world merchant fraud detection. The goal of the task is to predict whether a merchant is involved in fraudulent activities based on their daily transactions. We collected 33,744,737 merchants' transactions made from April to June of 2020 from a mobile payment platform and extracted 182 daily transaction features for each merchant (e.g., payment volume, transaction number). Finally, 272,821 normal instances and 205 fraud instances from April and May were selected for training, while 160,695 normal instances and 106 fraud instances from June were used for testing. Since a subsequent on-the-spot investigation required for verification is time-consuming and labor-intensive, this application would require a fraud detection model that is both accurate and aimed at reducing the number of false positives as much as possible.

Table 3 summarizes the results obtained by the seven baselines and PIA-WAL. Note that since the proportion of anomalies and normal instances is extremely unbalanced (1:1516), AUPRC is considered to be more appropriate for evaluating the performance of the models [22]. Evidently, compared with the baselines, PIA-WAL achieves a substantial improvement of 2.3%-50% (with a small standard deviation) in AUPRC, which demonstrates the effectiveness of PIA-WAL in merchant fraud detection. Moreover, we also calculated threshold-specific metrics including TN (True Negatives), FN (False Negatives), FP (False Positives) and TP (True Positives). As Table 3 shows, GAN-based methods obtain high TPs, but tend to produce massive FPs (from 14,524 to 33,544). PIA-WAL obtains comparable TP values when compared to iForest, ADOA and

DevNet. However, iForest and ADOA incur high FPs, 6,416 and 6,782, respectively. DevNet achieves the lowest FPs among the baselines, but the number of FPs is two times larger than the one obtained by PIA-WAL. Note that Deep SAD does not provide a comprehensive threshold selection strategy and thus its threshold-specific metric values are omitted from Table 3. Finally, the proposed PIA-WAL is able to significantly reduce the number of false positives while maintaining the anomaly detection accuracy in a real-world application.

7 Conclusion

We introduce a novel end-to-end anomaly detection model, PIA-WAL, which utilizes a few labeled anomalies to guide an adversarial training process. The main contribution is the generation of peripheral normal instances as supplements, which allows for PIA-WAL to learn a more representative description of the normal class. PIA-WAL achieves significant lifts over seven state-of-the-art methods on four public datasets. Even when the anomaly contamination is high or the number of labeled anomalies is low, PIA-WAL still obtains satisfactory performance. When applied to a real merchant fraud detection application, PIA-WAL can indeed reduce the number of false positives and maintain the anomaly detection accuracy, thus avoiding unnecessary labor and material resources.

Acknowledgements. This research was supported in part by NSFC grant 61902127 and Natural Science Foundation of Shanghai 19ZR1415700.

References

1. Akcay, S., Atapour-Abarghouei, A., Breckon, T.P.: Ganomaly: Semi-supervised anomaly detection via adversarial training. In: ACCV. pp. 622–637 (2018)
2. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein generative adversarial networks. In: ICML. pp. 214–223 (2017)
3. Audibert, J., Michiardi, P., Guyard, F., Marti, S., Zuluaga, M.A.: Usad: Unsupervised anomaly detection on multivariate time series. In: SIGKDD. pp. 3395–3404 (2020)
4. Bay, S.D., Schwabacher, M.: Mining distance-based outliers in near linear time with randomization and a simple pruning rule. In: SIGKDD. pp. 29–38 (2003)
5. Bergmann, P., Fauser, M., Sattlegger, D., Steger, C.: Mvtec ad–a comprehensive real-world dataset for unsupervised anomaly detection. In: CVPR. pp. 9592–9600 (2019)
6. Boukerche, A., Zheng, L., Alfandi, O.: Outlier detection: Methods, models, and classification. ACM Computing Surveys pp. 1–37 (2020)
7. Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J.: Lof: identifying density-based local outliers. In: SIGMOD. pp. 93–104 (2000)
8. Chen, J., Sathe, S., Aggarwal, C., Turaga, D.: Outlier detection with autoencoder ensembles. In: SDM. pp. 90–98 (2017)
9. Dal Pozzolo, A., Boracchi, G., Caelen, O., Alippi, C., Bontempi, G.: Credit card fraud detection: a realistic modeling and a novel learning strategy. IEEE Transactions on Neural Networks and Learning Systems pp. 3784–3797 (2017)

10. Di Mattia, F., Galeone, P., De Simoni, M., Ghelfi, E.: A survey on gans for anomaly detection. arXiv preprint arXiv:1906.11632 (2019)
11. Dua, D., Graff, C.: UCI machine learning repository (2017), http://archive.ics.uci.edu/ml
12. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks. NIPS p. 2672–2680 (2014)
13. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C.: Improved training of wasserstein gans. In: NIPS. pp. 5767–5777 (2017)
14. Kriegel, H.P., Kroger, P., Schubert, E., Zimek, A.: Interpreting and unifying outlier scores. In: SDM. pp. 13–24. SIAM (2011)
15. Liu, F.T., Ting, K.M., Zhou, Z.H.: Isolation-based anomaly detection. TKDD pp. 1–39 (2012)
16. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: ICCV (December 2015)
17. Moustafa, N., Slay, J.: Unsw-nb15: a comprehensive data set for network intrusion detection systems. In: MilCIS. pp. 1–6 (2015)
18. Ngo, P.C., Winarto, A.A., Kou, C.K.L., Park, S., Akram, F., Lee, H.K.: Fence gan: Towards better anomaly detection. In: ICTAI. pp. 141–148 (2019)
19. Pang, G., Cao, L., Chen, L., Liu, H.: Learning representations of ultrahigh-dimensional data for random distance-based outlier detection. In: SIGKDD. pp. 2041–2050 (2018)
20. Pang, G., van den Hengel, A., Shen, C., Cao, L.: Toward deep supervised anomaly detection: Reinforcement learning from partially labeled anomaly data. In: SIGKDD. pp. 1298–1308 (2021)
21. Pang, G., Shen, C., Cao, L., Hengel, A.V.D.: Deep learning for anomaly detection: A review. ACM Computing Surveys pp. 1–38 (2021)
22. Pang, G., Shen, C., van den Hengel, A.: Deep anomaly detection with deviation networks. In: SIGKDD. pp. 353–362 (2019)
23. Perera, P., Nallapati, R., Xiang, B.: Ocgan: One-class novelty detection using gans with constrained latent representations. In: CVPR. pp. 2898–2906 (2019)
24. Ruff, L., Vandermeulen, R.A., Görnitz, N., Binder, A., Müller, E., Müller, K.R., Kloft, M.: Deep semi-supervised anomaly detection. ICLR (2020)
25. Ruff, L., Vandermeulen, R.A., Görnitz, N., Deecke, L., Siddiqui, S.A., Binder, A., Müller, E., Kloft, M.: Deep one-class classification. In: ICML. pp. 4393–4402 (2018)
26. Schlegl, T., Seeböck, P., Waldstein, S.M., Schmidt-Erfurth, U., Langs, G.: Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In: IPMI. pp. 146–157 (2017)
27. Seeböck, P., Waldstein, S.M., Klimscha, S., Bogunovic, H., Schlegl, T., Gerendas, B.S., Donner, R., Schmidt-Erfurth, U., Langs, G.: Unsupervised identification of disease marker candidates in retinal oct imaging data. IEEE Transactions on Medical Imaging pp. 1037–1047 (2018)
28. Villani, C.: Optimal transport: old and new. In: Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]. vol. 338. Springer (2009)
29. Wang, H., Pang, G., Shen, C., Ma, C.: Unsupervised representation learning by predicting random distances. In: IJCAI. pp. 2950–2956 (2020)
30. Woolson, R.: Wilcoxon signed-rank test. Wiley encyclopedia of clinical trials pp. 1–3 (2007)
31. Zenati, H., Foo, C.S., Lecouat, B., Manek, G., Chandrasekhar, V.R.: Efficient gan-based anomaly detection. ICLR (2018)

32. Zenati, H., Romain, M., Foo, C.S., Lecouat, B., Chandrasekhar, V.: Adversarially learned anomaly detection. In: ICDM. pp. 727–736 (2018)
33. Zhang, Y.L., Li, L., Zhou, J., Li, X., Zhou, Z.H.: Anomaly detection with partially observed anomalies. In: WWW. pp. 639–646 (2018)
34. Zheng, P., Yuan, S., Wu, X., Li, J., Lu, A.: One-class adversarial nets for fraud detection. In: AAAI. pp. 1286–1293 (2019)
35. Zhou, C., Paffenroth, R.C.: Anomaly detection with robust deep autoencoders. In: SIGKDD. pp. 665–674 (2017)
36. 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: ICLR (2018)