Deep Weakly-supervised Anomaly Detection

Guansong Pang*
Singapore Management University
Singapore, Singapore
pangguansong@gmail.com

Chunhua Shen
Zhejiang University
Hangzhou, China
chhshen@gmail.com

Huidong Jin
Data61
Canberra, Australia
warren.jin@csiro.au

Anton van den Hengel
University of Adelaide
Adelaide, Australia
anton.vandenhengel@adelaide.edu.au

ABSTRACT

Recent semi-supervised anomaly detection methods that are trained using small labeled anomaly examples and large unlabeled data (mostly normal data) have shown largely improved performance over unsupervised methods. However, these methods often focus on fitting abnormalities illustrated by the given anomaly examples only (*i.e.*, seen anomalies), and consequently they fail to generalize to those that are not, *i.e.*, new types/classes of anomaly unseen during training. To detect both seen and unseen anomalies, we introduce a novel deep weakly-supervised approach, namely Pairwise Relation prediction Network (PReNet), that *learns pairwise relation features and anomaly scores* by predicting the relation of any two randomly sampled training instances, in which the pairwise relation can be anomaly-anomaly, anomaly-unlabeled, or unlabeled-unlabeled. Since unlabeled instances are mostly normal, the relation prediction enforces a joint learning of anomaly-anomaly, anomaly-normal, and normal-normal pairwise discriminative patterns, respectively. PReNet can then detect any seen/unseen abnormalities that fit the learned pairwise abnormal patterns, or deviate from the normal patterns. Further, this pairwise approach also seamlessly and significantly augments the training anomaly data. Empirical results on 12 real-world datasets show that PReNet significantly outperforms nine competing methods in detecting seen and unseen anomalies. We also theoretically and empirically justify the robustness of our model w.r.t. anomaly contamination in the unlabeled data. The code is available at https://github.com/mala-lab/PReNet.

CCS CONCEPTS

• **Computing methodologies** → **Anomaly detection**; **Neural networks**.

KEYWORDS

Anomaly Detection, Deep Learning, Intrusion Detection

*Corresponding author: G. Pang

KDD '23, August 6–10, 2023, Long Beach, CA, USA.

ACM ISBN 979-8-4007-0103-0/23/08…$15.00
https://doi.org/10.1145/3580305.3599302

ACM Reference Format:
Guansong Pang, Chunhua Shen, Huidong Jin, and Anton van den Hengel. 2023. Deep Weakly-supervised Anomaly Detection. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23), August 6–10, 2023, Long Beach, CA, USA.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3580305.3599302

1 INTRODUCTION

Anomaly detection (AD) aims at identifying exceptional data instances that deviate significantly from the majority of data. It is of critical practical importance due to its broad applications in defensing against cyber-crimes (*e.g.*, network intrusions), user misbehavior (*e.g.*, fraudulent user accounts/reviews), web advertising abuses, and adverse drug reactions, to name a few [1, 17, 18]. Numerous AD methods have been introduced, most of which are unsupervised methods working on entirely unlabeled (mostly normal data) data [1, 32]. The popularity of the unsupervised methods is mainly because they avoid the significant cost of manually labeling large-scale anomaly data, which is required to support fully-supervised approaches. However, they operate without knowing what true anomalies look like, and as a result, they identify many noisy or uninteresting isolated data instances as anomalies, leading to high detection errors.

Recent semi-supervised AD methods [9, 30, 31, 33, 43, 64] aim to bridge the gap between supervised and unsupervised AD by utilizing a limited number of anomaly examples to train anomaly-informed detection models. This line of research is motivated by the fact that a small set of labeled anomaly examples (*e.g.*, some successfully detected anomalous events) can often be made available with a small cost in real-world applications. These labeled anomalies provide a strong indication of the anomalies of interest and can substantially enhance the detection accuracy [9, 30, 31, 33, 43, 64]. However, anomalies are unknown abnormal events, so the labeled anomaly examples typically provides only an *incomplete illustration of anomalies*. The current methods focus on fitting the abnormalities illustrated by the small anomaly examples (*i.e.*, **seen anomalies**); they fail to generalize to those that are not, *i.e.*, new types/classes of anomaly unseen during training (**unseen anomalies**), such as zero-day attacks [34, 56] and novel defects/planet surfaces [8, 20, 31]. Also, their performance in detecting seen anomalies is restricted due to the lack of large training anomaly data.

To tackle these issues, this work considers the problem of **weakly-supervised AD**, or alternatively open-set supervised AD, that aims to detect both seen and unseen anomalies given an incomplete

illustration of anomaly classes seen during training. To this end, we introduce a novel deep AD approach, namely Pairwise Relation prediction Network (**PReNet**), that learns pairwise relation features and anomaly scores by predicting the relation of any two training instances randomly sampled from the small anomaly data and the unlabeled data, in which the pairwise relation labels can be *anomaly-anomaly, anomaly-unlabeled, or unlabeled-unlabeled*. During inference, a test instance is considered as an anomaly, if it fits well to the first two types of pairs, or deviates from the last pair type, when paired with a random training instance. In essence, our approach unifies the relation prediction and anomaly scoring, and learns to assign larger prediction scores (*i.e.*, anomaly scores) to the instance pairs that contain anomalies than the other instance pairs.

Our key insight is that since the unlabeled data is often mostly normal, the pairwise class labels offer rich three-way pairwise relation information that supports a joint learning of diverse discriminative patterns, including *anomaly-anomaly, anomaly-normal, and normal-normal pairwise feature patterns*, avoiding the fitting of the seen abnormalities only. Our approach can then detect any seen/unseen abnormalities that fit the learned pairwise abnormal patterns, or deviate from the normal patterns. Further, the pairwise relation formulation seamlessly generates large-scale anomaly-informed surrogate class labels, *i.e.*, the anomaly-anomaly and anomaly-unlabeled labels vs unlabeled-unlabeled. This significantly extends the training anomaly data, supporting effective training of a generalized detection model with the limited labeled data.

In summary, this work makes four main contributions:

- **Problem and Approach**. We consider the under-explored yet crucial problem – weakly-supervised anomaly detection – and propose a novel pairwise relation learning approach PReNet to address the problem. PReNet learns diverse discriminative pairwise relation features, offering more generalized detection models than existing methods.
- **Detection Model**. PReNet is instantiated to a novel detection model that learns pairwise anomaly scores by minimizing a three-way prediction loss using a relation neural network. The model is trained with the support of significantly augmented pairwise anomaly data, enabling effective training of a generalized detection model.
- **Robustness**. We theoretically and empirically show that PReNet can effectively leverage the large unlabeled data while being tolerant to anomaly contamination.
- **Large Empirical Support**. Our empirical results on 12 real-world datasets show that PReNet (i) significantly outperforms nine state-of-the-art (SOTA) competing methods in detecting seen and unseen anomalies, and (ii) obtains substantially better sample efficiency, *e.g.*, it requires 50%-87.5% less labeled anomaly data to perform comparably well to, or better than, the best competing models.

2 RELATED WORK

Toward Supervised Anomaly Detection. Previous semi-supervised AD methods [4, 13, 15, 29, 50] focus on leveraging labeled normal instances to learn patterns of the normal class. Since a small amount of anomaly data is often available in many real-world applications, recent semi-supervised methods [9, 27, 30, 43, 52, 64, 65] are dedicated to utilizing small labeled anomaly data to learn anomaly detectors, *e.g.*, label propagation [27, 52, 53], representation learning [30, 43, 67], classification models [64], or newly proposed loss functions [9, 33], and they show that these limited labeled anomalies can substantially improve the detection accuracy. Among them, DevNet [31, 33] and Deep SAD (DSAD) [25, 43] are two most relevant methods, which achieve impressive detection performance by fitting a Gaussian prior-driven anomaly score distribution and a one-class hypersphere, respectively. However, they are prone to overfitting the given anomaly examples due to the lack of proper regularization and large-scale, diversified training anomaly samples. To address this issue, weakly-supervised AD [16, 31, 34, 68] and open-set supervised AD [8] tasks are recently introduced, aiming to detect both seen and unseen anomalies. We follow this line and introduce a novel pairwise relation learning approach.

This research line is also relevant to few-shot learning [11, 21, 47, 51, 54, 57] and positive and unlabeled data (PU) learning [2, 5, 10, 19, 23, 26, 44] due to the availability of the limited labeled positive instances (anomalies), but they are very different in that these two areas assume that the few labeled instances share the same intrinsic class structure as the other instances within the same class (*i.e.*, the anomaly class), whereas the seen anomalies and the unseen anomalies may have completely different class structures.

Deep Anomaly Detection. Traditional AD approaches are often ineffective in high-dimensional or non-linear separable data due to the curse of dimensionality and the deficiency in capturing the non-linear relations [1, 32, 62]. Deep AD has shown promising results in handling those complex data, of which most methods are based on pre-trained features [40, 41], or features learned by using autoencoder- [6, 12, 36, 61] or generative adversarial network- [22, 45, 60, 63] based objectives. One issue with these methods is that these feature representations are not primarily optimized to detect anomalies. Some very recent methods [30, 42, 66, 70] address this issue by learning representations tailored for specific anomaly measures, *e.g.*, cluster membership-based measure in [49, 70], distance-based measure in [30, 55] and one-class classification-based measure in [7, 14, 42, 43, 48, 66]. However, they still focus on optimizing the feature representations. By contrast, our model unifies representation learning and anomaly scoring into one pipeline to directly optimize anomaly scores, yielding more optimized anomaly scores. Further, these methods overwhelmingly focus on unsupervised/semi-supervised settings where detection models are trained on unlabeled data or exclusively normal data, which fail to utilize the valuable labeled anomaly data as available in many real-world applications. Some recent studies such as DevNet [9, 31, 33] directly optimize the anomaly scores via a loss function called deviation loss [31, 33]. Our method instead uses a novel formulation of pairwise relation prediction to achieve the goal, which shows to be significantly better than the deviation loss.

Additionally, our relation prediction is formulated as a weakly-supervised three-way ordinal regression task which is different from [35] in both of the targeted problem and the approach taken since [35] uses self-trained ordinal regression for unsupervised AD. Also, anomaly contamination estimation [37–39] can help estimate the proportion of anomalies in the unlabeled data, which may be used as a prior for empowering weakly-supervised AD.

3 THE PROPOSED APPROACH

3.1 Overview of Our Approach PReNet

3.1.1 Problem Statement. Given a training dataset $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N, \mathbf{x}_{N+1}, \cdots, \mathbf{x}_{N+K}\}$, with $\mathbf{x}_i \in \mathbb{R}^D$, where $\mathcal{U} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$ is a large unlabeled dataset and $\mathcal{A} = \{\mathbf{x}_{N+1}, \mathbf{x}_{N+2}, \cdots, \mathbf{x}_{N+K}\}$ ($K \ll N$) is a small set of labeled anomaly examples that often do not illustrate every possible class of anomaly, our goal is to learn a scoring function $\phi : \mathcal{X} \rightarrow \mathbb{R}$ that assigns anomaly scores to data instances in a way that we have $\phi(\mathbf{x}_i) > \phi(\mathbf{x}_j)$ if $\mathbf{x}_i$ is an anomaly (despite it is a seen or unseen anomaly) and $\mathbf{x}_j$ is a normal instance.

3.1.2 Anomaly-informed Pairwise Relation Prediction. In our proposed approach PReNet, we formulate the problem as a pairwise relation prediction-based anomaly score learning, in which we learn to discriminate three types of random instance pairs, including anomaly-anomaly pairs, anomaly-unlabeled pairs, unlabeled-unlabeled pairs. The formulation unifies the relation prediction and anomaly scoring, and helps enforce the model to assign substantially larger prediction scores (*i.e.*, anomaly scores) to the instance pairs that contain anomalies than the other instance pairs. By doing so, the model learns diverse discriminative pairwise relation features embedded in the three-way pairwise interaction data. This way helps alleviate the overfitting of the seen anomalies as the model is regularized by simultaneously learning a variety of pairwise normality/abnormality patterns, rather than the seen abnormalities only. Further, the pairwise relation labels generate significantly more labeled training data than the original data, offering sufficiently large surrogate labeled data to train a generalized detection model.

Specifically, as shown in Fig. 1, our approach consists of two main modules: *anomaly-informed random instance pairing* and *pairwise relation-based anomaly score learning*. The first module generates an instance pair dataset $\mathcal{P} = \left\{\left(\mathbf{x}_i, \mathbf{x}_j, y_{\{\mathbf{x}_i,\mathbf{x}_j\}}\right) \mid \mathbf{x}_i, \mathbf{x}_j \in \mathcal{X} \text{ and } y_{\{\mathbf{x}_i,\mathbf{x}_j\}} \in \mathbb{N}\right\}$, where each pair $\{\mathbf{x}_i, \mathbf{x}_j\}$ has one of the three pairwise relations: $C_{\{\mathbf{a},\mathbf{a}\}}$, $C_{\{\mathbf{a},\mathbf{u}\}}$ and $C_{\{\mathbf{u},\mathbf{u}\}}$ ($\mathbf{a} \in \mathcal{A}$ and $\mathbf{u} \in \mathcal{U}$) and $\mathbf{y} \in \mathbb{N}^{|\mathcal{P}|}$ is an ordinal class feature with *decreasing* value assignments to the respective $C_{\{\mathbf{a},\mathbf{a}\}}$, $C_{\{\mathbf{a},\mathbf{u}\}}$ and $C_{\{\mathbf{u},\mathbf{u}\}}$ pairs, *i.e.*, $y_{\{\mathbf{a},\mathbf{a}\}} > y_{\{\mathbf{a},\mathbf{u}\}} > y_{\{\mathbf{u},\mathbf{u}\}}$. These pairwise labels are set to be ordinal values to enable the anomaly score learning module $\phi : \mathcal{P} \rightarrow \mathbb{R}$, which can be treated as jointly learning a feature learner ψ and a relation (anomaly score) learner η. ϕ is trained in an end-to-end manner to learn the pairwise anomaly scores using $\mathcal{P}$.

3.2 The Instantiated Model

The two modules of PReNet are specified as follows.

3.2.1 Anomaly-informed Random Instance Pairing. In this module, PReNet generates large-scale instance pairs with surrogate class labels to provide large labeled data for training subsequent pairwise relation prediction models. Specifically, instance pairs are created with instances randomly sampled from the small anomaly set $\mathcal{A}$ and the large unlabeled dataset $\mathcal{U}$. A pairwise class label is then assigned to each instance pair, such that $y_{\{\mathbf{a},\mathbf{a}\}} = c_1$, $y_{\{\mathbf{a},\mathbf{u}\}} = c_2$, $y_{\{\mathbf{u},\mathbf{u}\}} = c_3$ and $c_1 > c_2 > c_3 \geq 0$. By doing so, we efficiently synthesize $\mathcal{A}$ and $\mathcal{U}$ to produce a large labeled dataset $\mathcal{P} = \left\{\left(\mathbf{x}_i, \mathbf{x}_j, y_{\{\mathbf{x}_i,\mathbf{x}_j\}}\right) \mid \mathbf{x}_i, \mathbf{x}_j \in \mathcal{X} \text{ and } y_{\{\mathbf{x}_i,\mathbf{x}_j\}} \in \mathbb{N}\right\}$.

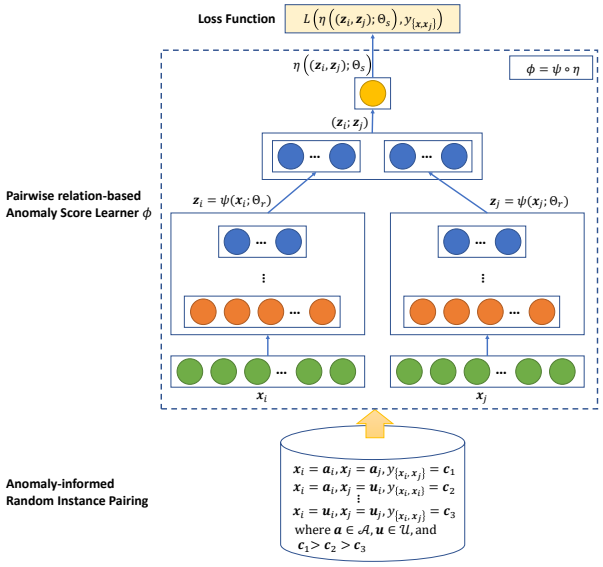

Figure 1: Overview of PReNet. It takes anomaly-anomaly, anomaly-unlabeled, and unlabeled-unlabeled instance pairs as input and learns pairwise anomaly scores by discriminating these three types of linear pairwise interactions.

The resulting $\mathcal{P}$ contains critical information for discriminating anomalies from normal instances. This is because $y_{\{\mathbf{a},\mathbf{a}\}}$, $y_{\{\mathbf{a},\mathbf{u}\}}$ and $y_{\{\mathbf{u},\mathbf{u}\}}$ are approximately anomaly-anomaly, anomaly-normal and normal-normal pairs, respectively, as $\mathcal{U}$ is typically dominated by normal instances (per definition of anomaly [1, 4]). A few $y_{\{\mathbf{a},\mathbf{u}\}}$ and $y_{\{\mathbf{u},\mathbf{u}\}}$ pairs may be noisy pairs due to anomaly contamination in $\mathcal{U}$, but we show that PReNet is robust to these noisy pairs

3.2.2 Pairwise Relation-based Anomaly Score Learning. A pairwise anomaly score learner $\phi : \mathcal{P} \rightarrow \mathbb{R}$ is then introduced to take $\mathcal{P}$ as input to learn the anomaly scores of instance pairs. Let $\mathcal{Z} \in \mathbb{R}^M$ be an intermediate representation space, we define a two-stream anomaly scoring network $\phi\left((\cdot,\cdot);\Theta\right) : \mathcal{P} \rightarrow \mathbb{R}$ as a sequential combination of a feature learner $\psi(\cdot;\Theta_r) : \mathcal{X} \rightarrow \mathcal{Z}$ and an anomaly scoring function $\eta\left((\cdot,\cdot);\Theta_s\right) : (\mathcal{Z},\mathcal{Z}) \rightarrow \mathbb{R}$, where $\Theta = \{\Theta_r, \Theta_s\}$. Specifically, $\psi(\cdot;\Theta_r)$ is a neural *feature learner* with $H \in \mathbb{N}$ hidden layers and the weight parameters Θ_r.

$$\mathbf{z} = \psi(\mathbf{x};\Theta_r), \tag{1}$$

where $\mathbf{x} \in \mathcal{X}$ and $\mathbf{z} \in \mathcal{Z}$. We further specify $\eta\left((\cdot,\cdot);\Theta_s\right)$ as an *anomaly score learner* that uses a fully connected layer to learn linear pairwise relation features and the anomaly scores, taking the concatenation of the intermediate representations of each pair – $\mathbf{z}_i$ and $\mathbf{z}_j$ – as input:

$$\eta\left((\mathbf{z}_i,\mathbf{z}_j);\Theta_s\right) = \sum_{k=1}^{M} w_k^o z_{ik} + \sum_{l=1}^{M} w_{M+l}^o z_{jl} + w_{2M+1}^o, \tag{2}$$

where $\mathbf{z} \in \mathcal{Z}$ and $\Theta_s = \{\mathbf{w}^o\}$ in which $\{w_1^o, w_2^o, \cdots, w_{2M}^o\}$ are weight parameters and w_{2M+1}^o is a bias term. As shown in Fig. 1, PReNet uses a two-stream network with the shared weight parameters Θ_r to learn the representations $\mathbf{z}_i$ and $\mathbf{z}_j$. Thus, $\phi\left((\cdot,\cdot);\Theta\right)$ can

be formally represented as

$$\phi\big((\mathbf{x}_i, \mathbf{x}_j); \Theta\big) = \eta\Big(\big(\psi(\mathbf{x}_i; \Theta_r), \psi(\mathbf{x}_j; \Theta_r)\big); \Theta_s\Big), \tag{3}$$

which can be trained in an end-to-end fashion. Note that the pairwise relation learned in Eqn. (2) is a simple linear relation; learning more complex relations can be done by adding more layers with non-linear activation on top of the concatenated features, but it does not show clear advantages in our setting (see Table 4).

PReNet then uses the pairwise ordinal class labels to optimize the pairwise relation-based anomaly scores. Particularly, it minimizes the difference between the prediction scores and the ordinal labels $y_{\{\mathbf{a},\mathbf{a}\}}$, $y_{\{\mathbf{a},\mathbf{u}\}}$ and $y_{\{\mathbf{u},\mathbf{u}\}}$. It is equivalent to learning to assign larger prediction scores to the anomaly-related (*i.e.*, anomaly-anomaly and anomaly-unlabeled) instance pairs than the unlabeled-unlabeled pairs. Our loss is defined as below to guide the optimization:

$$L\Big(\phi\big((\mathbf{x}_i, \mathbf{x}_j); \Theta\big), y_{\{\mathbf{x}_i,\mathbf{x}_j\}}\Big) = \Big|y_{\{\mathbf{x}_i,\mathbf{x}_j\}} - \phi\big((\mathbf{x}_i, \mathbf{x}_j); \Theta\big)\Big|. \tag{4}$$

The three-class labels $y_{\{\mathbf{a},\mathbf{a}\}} = 8$, $y_{\{\mathbf{a},\mathbf{u}\}} = 4$ and $y_{\{\mathbf{u},\mathbf{u}\}} = 0$ are used by default to enforce a large margin among the anomaly scores of the three types of instance pairs. PReNet also works well with other value assignments as long as there are reasonably large margins among the ordinal labels (see Sec. 5.6). Lastly, PReNet is trained via:

$$\arg\min_{\Theta} \frac{1}{|\mathcal{B}|} \sum_{(\mathbf{x}_i,\mathbf{x}_j,y_{\{\mathbf{x}_i,\mathbf{x}_j\}})\in\mathcal{B}} \Big|y_{\{\mathbf{x}_i,\mathbf{x}_j\}} - \phi\big((\mathbf{x}_i, \mathbf{x}_j); \Theta\big)\Big| + \lambda R(\Theta), \tag{5}$$

where $\mathcal{B}$ is a sample batch from $\mathcal{P}$ and $R(\Theta)$ is a regularization term with hyperparameter λ. For each batch, $\frac{|\mathcal{B}|}{2}$ instance pairs are sampled from the $C_{\{\mathbf{u},\mathbf{u}\}}$ class and $\frac{|\mathcal{B}|}{4}$ instance pairs are respectively sampled from the $C_{\{\mathbf{a},\mathbf{a}\}}$ and $C_{\{\mathbf{a},\mathbf{u}\}}$ classes. This is equivalent to oversampling the two anomaly-related classes, $C_{\{\mathbf{a},\mathbf{a}\}}$ and $C_{\{\mathbf{a},\mathbf{u}\}}$, to avoid bias toward the $C_{\{\mathbf{u},\mathbf{u}\}}$ class due to the class imbalance.

3.3 Anomaly Detection Using PReNet

Training. Algorithm 1 presents the procedure of training PReNet. Step 1 first extends the data $\mathcal{X}$ into a set of instance pairs with ordinal class labels, $\mathcal{P}$. After a uniform *Glorot* weight initialization in Step 2, PReNet performs stochastic gradient descent (SGD) based optimization to learn Θ in Steps 3-9 and obtains the optimized ϕ in Step 10. Particularly, stratified random sampling is used in Step 5 to ensure the sample balance of the three classes in $\mathcal{B}$, as discussed in Sec. 3.2.2. Step 6 performs the forward propagation of the network and computes the loss. Step 7 then uses the loss to perform gradient descent steps.

Inference. During inference, given a test instance $\mathbf{x}_k$, PReNet first pairs it with data instances randomly sampled from $\mathcal{A}$ and $\mathcal{U}$, and then defines its anomaly score as

$$s_{\mathbf{x}_k} = \frac{1}{2E}\left[\sum_{i=1}^{E} \phi\big(\,(\mathbf{a}_i, \mathbf{x}_k)\,; \Theta^*\big) + \sum_{j=1}^{E} \phi\big((\mathbf{x}_k, \mathbf{u}_j); \Theta^*\big)\right], \tag{6}$$

where Θ^* are the parameters of a trained ϕ, and $\mathbf{a}_i$ and $\mathbf{u}_j$ are randomly sampled from the respective $\mathcal{A}$ and $\mathcal{U}$. $s_{\mathbf{x}_k}$ can be interpreted as an ensemble of the anomaly scores of a set of $\mathbf{x}_k$-oriented pairs. Due to the loss in Eqn. (4), $s_{\mathbf{x}_k}$ is optimized to be greater than $s_{\mathbf{x}'_k}$ given $\mathbf{x}_k$ is an anomaly and $\mathbf{x}'_k$ is a normal instance. PReNet can perform stably with a sufficiently large E due to the law of large numbers ($E = 30$ is used by default; see Sec. 5.6 for other results).

Algorithm 1 *Training PReNet*

Input: $\mathcal{X} \in \mathbb{R}^D$ with $\mathcal{X} = \mathcal{U} \cup \mathcal{A}$ and $\emptyset = \mathcal{U} \cap \mathcal{A}$
Output: $\phi : (\mathcal{X}, \mathcal{X}) \rightarrow \mathbb{R}$ - an anomaly score mapping
1: $\mathcal{P} \leftarrow$ Augment the training data with $\mathcal{U}$ and $\mathcal{A}$
2: Randomly initialize Θ
3: **for** $i = 1$ to *n_epochs* **do**
4: **for** $j = 1$ to *n_batches* **do**
5: $\mathcal{B} \leftarrow$ Randomly sample b data instance pairs from $\mathcal{P}$
6: $loss \leftarrow \frac{1}{b}\sum_{(\mathbf{x}_i,\mathbf{x}_j,y_{\{\mathbf{x}_i,\mathbf{x}_j\}})\in\mathcal{B}} \Big|y_{\{\mathbf{x}_i,\mathbf{x}_j\}} - \phi\big((\mathbf{x}_i, \mathbf{x}_j); \Theta\big)\Big| + \lambda R(\Theta)$
7: Perform a gradient descent step w.r.t. the parameters in Θ
8: **end for**
9: **end for**
10: **return** ϕ

4 THEORETICAL ANALYSIS

4.1 Pairwise Relation Feature Learning

The random instance pairing module seamlessly leverages the two instance sets $\mathcal{A}$ and $\mathcal{U}$ to create large-scale proxy class labels to support pairwise relation feature learning. That is, the sample size of the training data theoretically increases from $N + K$ in the original data space to $(N + K)^2$ for the pairwise relation learning, including K^2 of $C_{\{\mathbf{a},\mathbf{a}\}}$ pairs, $2K \times N$ of $C_{\{\mathbf{a},\mathbf{u}\}}$ pairs and N^2 of $C_{\{\mathbf{u},\mathbf{u}\}}$ pairs (note that as set notion is used, $\{\mathbf{a}, \mathbf{u}\} = \{\mathbf{u}, \mathbf{a}\}$). Such a large size helps build up the generalizability and then the performance of our detector. Note that PReNet uses a shared-weight two-stream network in $\psi(\cdot; \Theta_r)$, so the feature learning is still optimized on the $\mathcal{X}$ data space rather than the higher-order pairwise $\mathcal{P}$ space. This trick well supports the scale-up of the training sample size while adding no extra model complexity. Further, the relation learning in Eqn. (5) enforces PReNet to discriminate the representations of anomaly-anomaly, anomaly-normal and normal-normal pairwise interactions (as $\mathcal{U}$ contains mostly normal data). This results in a joint learning of diverse patterns of abnormality, normality, and their interactions, avoiding the exclusive fitting of the seen abnormalities that may consequently overfit the seen abnormalities and fail to generalize to unseen abnormalities.

4.2 Robust Anomaly Score Learning

This section analyzes the robustness of PReNet to ϵ-contamination in $\mathcal{U}$, where ϵ is the proportion of true anomalies in $\mathcal{U}$. Per the definition of anomaly, ϵ is typically small, *e.g.*, $< 2\%$. From the three-way modeling of anomaly-anomaly, anomaly-normal and normal-normal interactions, we can obtain the expectation of the pairwise relation proportions in each batch $\mathcal{B}$ based on uniformly random sampling. Let a_T and n_T indicate true anomaly and true normal instances, respectively, we can then have the probability expectation of each type of the interactions in $\mathcal{P}$ in Table 1.

Considering the $\frac{1}{4}$,$\frac{1}{4}$, and $\frac{1}{2}$ sampling probability of anomaly and unlabeled pairs in $\mathcal{B}$, there are $\frac{1}{4} + \frac{1}{4}\epsilon + \frac{1}{2}\epsilon^2$ from true anomaly-anomaly pairwise relations, and $\frac{1}{4} + \frac{3}{4}\epsilon - \epsilon^2$ from true anomaly-normal pairwise relations, and $\frac{1}{2} - \epsilon + \frac{1}{2}\epsilon^2$ from normal-normal

pairwise relations. Their true expectation values when all the unlabeled cases are not true anomalies, are $\frac{1}{4}$, $\frac{1}{4}$, and $\frac{1}{2}$ respectively. Thus, a small percentage of the pairwise relations, $\left|\frac{1}{4}\epsilon + \frac{1}{2}\epsilon^2\right| + \left|\frac{3}{4}\epsilon - \epsilon^2\right| + \left|-\epsilon + \frac{1}{2}\epsilon^2\right| = 2\epsilon - \epsilon^2$, would be expected to be noisy pairs. Considering the tolerance of the regression performance to about 5% outliers [46, 59], PReNet can perform well when ϵ is reasonably small, $\leq 2.5\%$, which can often be satisfied for real-world anomaly detection problems. On the other hand, from the regression analysis, we can derive the following theorem for the expectation for different types of pairwise relation and their anomaly scores.

Table 1: Probability expectation of pairwise interactions

	$\{a_T, a_T\}$	$\{a_T, n_T\}$	$\{n_T, n_T\}$
$\{\mathbf{a}, \mathbf{a}\}$	100%		
$\{\mathbf{a}, \mathbf{u}\}$	$1 * \epsilon$	$1 - \epsilon$	
$\{\mathbf{u}, \mathbf{u}\}$	$\epsilon * \epsilon$	$2\epsilon\,(1-\epsilon)$	$(1-\epsilon)\,(1-\epsilon)$

Theorem 4.1 (Robustness to Anomaly Contamination). *Let $\epsilon \geq 0$ be the anomaly contamination rate in $\mathcal{U}$, $y_{\{\mathbf{a},\mathbf{a}\}} = c_1$, $y_{\{\mathbf{a},\mathbf{u}\}} = c_2$, and $y_{\{\mathbf{u},\mathbf{u}\}} = c_3$ with $c_1 > c_2 > c_3 \geq 0$, then for a given test instance $\mathbf{x}_k$, we have $\mathbb{E}\left[s_{\mathbf{x}_k} | \mathbf{x}_k \text{ is an anomaly}\right] = \frac{c_1+c_2}{2}$, which is guaranteed to be greater than $\mathbb{E}\left[s_{\mathbf{x}_k} | \mathbf{x}_k \text{ is normal}\right] = \frac{c_2+c_3+\epsilon(c_1+c_2)}{2}$ for $\epsilon < \frac{c_1-c_3}{c_1+c_2}$.*

Proof. From the regression modeling, the expectation for the pairwise anomaly score for a true anomaly $\mathbf{a}_k$ is

$$\mathbb{E}\left[\phi\left((\mathbf{a}_i, \mathbf{a}_k); \Theta^*\right)\right] = c_1,$$
$$\mathbb{E}\left[\phi\left((\mathbf{a}_k, \mathbf{u}_j); \Theta^*\right)\right] = c_2,$$

The corresponding expectation for a normal data instance $\mathbf{n}_l$ is given by

$$\begin{aligned}\mathbb{E}\left[\phi\left((\mathbf{a}_i, \mathbf{n}_l); \Theta^*\right)\right] &= \mathbb{E}\left[\phi\left((\mathbf{a}_i, \mathbf{u}_j); \Theta^*\right)\right] + \epsilon\mathbb{E}\left[\phi\left((\mathbf{a}_i, \mathbf{a}_j); \Theta^*\right)\right] \\ &= c_2 + \epsilon c_1, \\ \mathbb{E}\left[\phi\left((\mathbf{n}_l, \mathbf{u}_j); \Theta^*\right)\right] &= \mathbb{E}\left[\phi\left((\mathbf{u}_i, \mathbf{u}_j); \Theta^*\right)\right] + \epsilon\mathbb{E}\left[\phi\left((\mathbf{a}_i, \mathbf{u}_j); \Theta^*\right)\right] \\ &= c_3 + \epsilon c_2.\end{aligned}$$

Thus, based on the anomaly scoring function in Eqn. (6):

$$s_{\mathbf{x}_k} = \frac{1}{2E}\left[\sum_{i=1}^{E}\phi\left((\mathbf{a}_i, \mathbf{x}_k); \Theta^*\right) + \sum_{j=1}^{E}\phi\left((\mathbf{x}_k, \mathbf{u}_j); \Theta^*\right)\right], \quad (7)$$

we have

$$\mathbb{E}\left[s_{\mathbf{x}_k} | \mathbf{x}_k \text{ is an anomaly}\right] = \frac{1}{2E}\left\{E \times \mathbb{E}\left[\phi\left((\mathbf{a}_i, \mathbf{a}_k); \Theta^*\right)\right] + E \times \mathbb{E}\left[\phi\left((\mathbf{a}_k, \mathbf{u}_j); \Theta^*\right)\right]\right\} = \frac{c_1 + c_2}{2}, \quad (8)$$

and

$$\mathbb{E}\left[s_{\mathbf{x}_k} | \mathbf{x}_k \text{ is normal}\right] = \frac{1}{2E}\left\{E \times \mathbb{E}\left[\phi\left((\mathbf{a}_i, \mathbf{n}_l); \Theta^*\right)\right] + E \times \mathbb{E}\left[\phi\left((\mathbf{n}_l, \mathbf{u}_j); \Theta^*\right)\right]\right\} = \frac{c_2 + c_3 + \epsilon(c_1 + c_2)}{2}. \quad (9)$$

Since $c_1 > c_2 > c_3 \geq 0$, $\mathbb{E}\left[s_{\mathbf{x}_k} | \mathbf{x}_k \text{ is an anomaly}\right]$ is guaranteed to be greater than $\mathbb{E}\left[s_{\mathbf{x}_k} | \mathbf{x}_k \text{ is normal}\right]$ for $\epsilon < \frac{c_1-c_3}{c_1+c_2}$. □

This theorem indicates that in PReNet a true anomaly is expected to have a larger anomaly score than normal instances when the contamination rate in $\mathcal{U}$ is not too large. That is, we have $\epsilon < \frac{2}{3}$ for our default setting: $c_1 = 8$, $c_2 = 4$ and $c_3 = 0$ (see Sec. 3.2.2), which is normally satisfied in real-world anomaly detection applications.

5 EXPERIMENTS

5.1 Datasets

Multidimensional (or tabular) data is ubiquitous in real-world applications, so we focus on this type of publicly available datasets[1]. To explicitly evaluate the performance of detecting seen/unseen anomalies, we have two groups of datasets from the literature [24, 28, 33], including 12 datasets used for the detection of seen anomalies and another 28 datasets used for detecting unseen anomalies:

Seen Anomaly Detection Datasets As shown in Table 2, 12 real-world datasets are used for the detection of seen anomalies, which are from diverse domains, *e.g.*, cyber-attack detection, fraud detection, and disease risk assessment. Each dataset contains 0.2%-15.0% anomalies of the same class. To replicate the real-world scenarios where we have a few labeled seen anomalies and large unlabeled data, we first have a stratified split of each dataset into two subsets, with 80% data as training data and the other 20% data as a holdup test set. Since the unlabeled data is often anomaly-contaminated, we then combine some randomly selected anomalies with the normal training instances to form the unlabeled data $\mathcal{U}$. We further randomly sample a limited number of anomalies from the anomaly class to form the labeled anomaly set $\mathcal{A}$.

Unseen Anomaly Detection Datasets The 28 datasets for detecting unseen anomalies are presented in Table 3. These datasets are derived from four intrusion attack datasets *dos*, *rec*, *fuz* and *bac* in Table 2[2], whose data instances are from the same data source and spanned by the same feature space. To guarantee that the anomalies in the test data are unseen during training, the anomaly class in one of these four datasets is held up for evaluation, while the anomalies in any combinations of the remaining three datasets are combined to form the pool of seen anomalies. We have 28 possible permutations under this setting, resulting in 28 datasets with different seen and/or unseen anomaly classes, as shown in Table 3. During training, $\mathcal{A}$ contains the anomalies sampled from the pool of seen anomalies, while the test data is composed of the held-up unseen anomaly classes and the normal instances in the test set.

5.2 Competing Methods and Their Settings

PReNet is compared with six state-of-the-art methods from several related areas, including semi-supervised anomaly detectors, DevNet [31, 33] and one-class classifier Deep SAD (DSAD) [43], highly class-imbalanced (few-shot) classifier FSNet [47] and its cost-sensitive variant cFSNet, and unsupervised anomaly detection methods iForest [24] and REPEN [30] (REPEN represents unsupervised detectors that have a component to easily utilize any available anomaly data to train their models). Similar to [30, 33], we found empirically that all deep methods using a multilayer perceptron

[1] See Appendix A.1 for more details about the used datasets.
[2] The other eight datasets cannot be used in evaluating unseen anomaly detection as they are from different data sources and contain only one anomaly class.

network architecture with one hidden layer perform better and more stably than using two or more hidden layers. Thus, following DevNet, one hidden layer with 20 neural units is used in all deep methods. The ReLu activation function $g(a) = max(0, a)$ is used. An ℓ_2-norm regularizer with the hyperparameter setting $\lambda = 0.01$ is applied to avoid overfitting. The RMSprop optimizer with the learning rate 0.001 is used. All deep detectors are trained using 50 epochs, with 20 batches per epoch. Similar to PReNet, oversampling is also applied to the labeled anomaly set $\mathcal{A}$ to well train the deep detection models of DevNet, REPEN, DSAD, FSNet and cFSNet. iForest with recommended settings [24] is used as a baseline here.

We also compare PReNet with XGBOD [65], PUMAD [19], and FEAWAD [68]. The comparison results are given in Appendix B due to space limitation.

5.3 Performance Evaluation Metrics

Two popular metrics – the Area Under Receiver Operating Characteristic Curve (AUC-ROC) and Area Under Precision-Recall Curve (AUC-PR) – are used. A larger AUC-ROC/AUC-PR reflects better performance. AUC-ROC that summarizes the curve of true positives against false positives often presents an overoptimistic view of the performance, whereas AUC-PR is more practical as it summarizes the precision and recall w.r.t. the anomaly class exclusively [3]. The reported results are averaged values over 10 independent runs. The paired *Wilcoxon* signed-rank [58] is used to examine the statistical significance of PReNet against its competing methods.

5.4 Detection of Seen Anomalies

Effectiveness of PReNet using small anomaly examples. We first evaluate PReNet on detecting seen anomalies in 12 real-world datasets. A consistent anomaly contamination rate and the same number of labeled anomalies are used across all datasets to gain insights into the performance in different real-life applications. Since anomalies are typically rare instances, the number of labeled anomalies available per data is set to 60, *i.e.*, $|\mathcal{A}| = 60$, and the anomaly contamination rate is set to 2% by default.

The results on the 12 datasets are shown in Table 2. In AUC-PR, PReNet performs substantially better than, or comparably well to, all competing methods across the 12 datasets. On average, PReNet improves all five competing methods by a large margin, *i.e.*, DevNet (3.7%), DSAD (19.6%), FSNet (54.0%), cFSNet (59.1%), REPEN (114.2%), and iForest (331.85%), which are all statistically significant at the 95%/99% confidence level. Particularly, compared to the top two contenders, PReNet significantly outperforms DevNet on six datasets, with improvement ranging from 3%-6% on *census*, *bac*, *news20* and *thyroid*, and up to 10%-20% on *campaign* and *w7a*; they perform comparably well on the rest of six datasets; PReNet performs significantly better than DSAD on eight datasets, achieving 20%-320% improvement on six datasets, including *donors*, *fuz*, *w7a*, *campaign*, *news20* and *thyroid*, and they perform comparably well on the rest of four datasets. In terms of AUC-ROC, PReNet outperforms DevNet at the 90% confidence level, and performs significantly better than DSAD (2.9%), FSNet (11.7%), cFSNet (13.1%), REPEN (10.6%) and iForest (40.7%) at the 95%/99% confidence level.

Using less/more labeled anomaly examples. We further examine PReNet by evaluating its performance w.r.t. different numbers of labeled anomalies, ranging from 15 to 120, with the contamination rate fixed to 2%. The AUC-PR results are shown in Fig. 2. iForest is omitted as it does not use labeled data. The results of all methods generally increases with labeled data size. However, The increased anomalies do not always help due to the heterogeneous anomalous behaviors taken by different anomalies. PReNet is more stable in the increasing trend. Consistent with the results in Table 2, PReNet still significantly outperforms its state-of-the-art competing methods with varying numbers of anomaly examples. Particularly, PReNet demonstrates the most sample-efficient learning capability. Impressively, PReNet can be trained with 50%-75% less labeled anomalies but achieves much better, or comparably good, AUC-PR than the best contender DevNet on multiple datasets like *dos*, *fuz*, *w7a* and *campaign*; and it is trained with 87.5% less labeled data while obtains substantially better performance than the second-best contender DSAD on *donors*, *w7a*, *campaign*, *news20* and *thyroid*. Similar observations apply to FSNet, cFSNet and REPEN.

5.5 Detection of Unseen Anomalies

Generalizing to unseen anomaly classes using small examples of seen anomaly classes. This section evaluates the detectors that are trained with only seen anomaly classes to detect unseen anomaly classes on the 28 datasets. Similarly to Sec. 5.4, the anomaly contamination rate of 2% and $|\mathcal{A}| = 60$ are used here. The results are presented in Table 3, in which iForest that is insensitive to the change is used as baseline. The AUC-PR results show that PReNet outperforms all the five competing methods by substantial margins on the 28 datasets. On average, PReNet improves DevNet by more than 11%, DSAD by 17%, FSNet by 30%, cFSNet by 20% and REPEN by 27%. It is impressive that, compared to the best competing method DevNet, PReNet achieves 20%-130% AUC-PR improvement on eight datasets, including 20%-40% improvement on '*rec* → *bac*', '*rec,fuz* → *dos*', '*rec, bac, fuz* → *dos*', '*bac, fuz* → *dos*', '*fuz* → *dos*', '*rec, bac, dos* → *fuz*', and over 100% improvement on '*rec, fuz* → *bac*' and '*fuz* → *bac*'. The improvement over the other four contenders is more substantial than that over DevNet. All these improvements are statistically significant at the 99% confidence level. PReNet gains similar superiority in AUC-ROC as well.

Detection of unseen anomaly classes with less/more seen anomalies. We examine this question on 14 unknown anomaly datasets where all methods work relatively well compared to the other 14 datasets. The AUC-PR results are presented in Fig. 3. It is interesting that most detectors gain improved performance in detecting unseen anomalies when more seen anomalies are given. This may be due to that more seen anomaly examples help better train the detection models, enabling a better anomaly discriminability. The superiority of PReNet here is consistent with that in Table 3. PReNet remains the most sample-efficient method, and can perform substantially better than the best competing methods even when the PReNet model is trained with 50%-87.5% less labeled data.

5.6 Further Analysis of PReNet

Robustness w.r.t. anomaly contamination. We investigate this robustness by using different anomaly contamination rates, {0%, 2%, 5%, 10%}, with $|\mathcal{A}| = 60$ fixed. The AUC-PR results for this experiment are presented in Fig. 4. PReNet performs generally

Table 2: Seen anomaly detection results. 'Size' is the data size. D is the dimension. '1M' denotes *news20* has 1,355,191 features.

Data Statistics			AUC-PR Results							AUC-ROC Results						
Data	**Size**	D	**PReNet**	**DevNet**	**DSAD**	**FSNet**	**cFSNet**	**REPEN**	**iForest**	**PReNet**	**DevNet**	**DSAD**	**FSNet**	**cFSNet**	**REPEN**	**iForest**
donors	619,326	10	**1.000**	0.997	0.806	0.995	0.998	0.520	0.222	**1.000**	**1.000**	0.993	0.999	1.000	0.976	0.875
census	299,285	500	**0.356**	0.345	0.330	0.197	0.197	0.173	0.078	**0.862**	0.861	0.858	0.759	0.759	0.822	0.634
fraud	284,807	29	0.689	0.693	**0.695**	0.157	0.169	0.678	0.261	0.980	**0.981**	0.980	0.776	0.762	0.974	0.946
celeba	202,599	39	**0.309**	0.306	0.306	0.103	0.112	0.166	0.060	0.960	**0.961**	0.958	0.855	0.855	0.899	0.686
dos	109,353	196	0.900	0.900	**0.910**	0.826	0.836	0.461	0.266	0.949	0.948	**0.956**	0.927	0.935	0.890	0.762
rec	106,987	196	0.767	0.760	**0.772**	0.650	0.679	0.302	0.132	0.966	0.962	**0.971**	0.926	0.936	0.829	0.534
fuz	96,000	196	**0.170**	0.136	0.133	0.146	0.146	0.075	0.039	**0.882**	0.873	0.875	0.858	0.858	0.789	0.548
bac	95,329	196	**0.890**	0.863	0.862	0.618	0.618	0.117	0.050	**0.976**	0.968	0.945	0.950	0.950	0.882	0.741
w7a	49,749	300	**0.496**	0.408	0.117	0.098	0.098	0.112	0.023	**0.883**	0.882	0.802	0.767	0.767	0.733	0.413
campaign	41,188	62	**0.470**	0.426	0.386	0.255	0.255	0.333	0.313	**0.880**	0.858	0.803	0.684	0.684	0.740	0.723
news20	10,523	1M	**0.652**	0.632	0.329	0.105	0.116	0.222	0.035	0.956	**0.960**	0.909	0.686	0.690	0.869	0.333
thyroid	7,200	21	**0.298**	0.280	0.205	0.149	0.175	0.108	0.144	**0.781**	0.767	0.713	0.590	0.594	0.613	0.679
	Average		**0.583**	0.562	0.488	0.358	0.367	0.272	0.135	**0.923**	0.918	0.897	0.815	0.816	0.835	0.656
	P-value		-	0.005	0.016	0.001	0.001	0.001	0.001	-	0.075	0.025	0.001	0.001	0.001	0.001

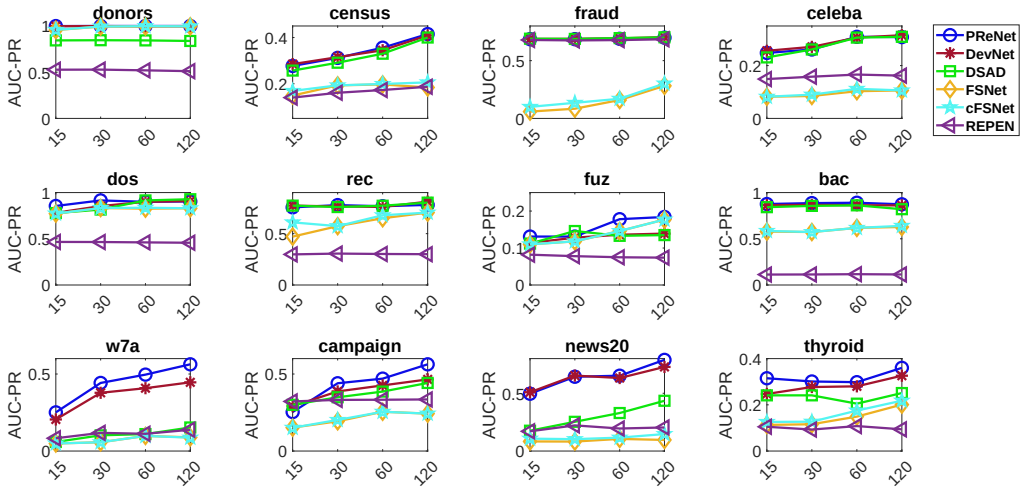

Figure 2: AUC-PR results of seen anomaly detection w.r.t. the number of labeled anomalies

stably on all datasets with the contamination rate below 10%, except *news20* that contains over one millions features and may therefore require better relation learning designs to achieve good robustness w.r.t. a large contamination rate.

Ablation study. In Table 4, PReNet is compared to its four ablated variants to evaluate the importance of each module:

- **Three-way relation modeling.** To examine the importance of learning diverse pairwise patterns via three-way relation modeling, we compare PReNet with its Two-way Relation Modeling (TRM) variant that learns two-way relations only, *i.e.*, to discriminate $y_{\{\mathbf{a},\mathbf{a}\}|\{\mathbf{a},\mathbf{u}\}}$ from $y_{\{\mathbf{u},\mathbf{u}\}}$. PReNet outperforms than TRM on nearly all datasets, resulting in large average improvement (5.6%). This indicates that learning more diverse patterns helps obtain better generalization.
- **Relation features.** Can PReNet perform similarly well by learning features of individual instances rather than pairwise relation features? The answer is negative. Compared to iPReNet in Table 4 that takes individual instances as input and learns to discriminate seen anomalies from unlabeled data, PReNet performs better by a large margin on 11 out of 12 datasets.
- **Feature learning layer ψ.** To evaluate the importance of intermediate feature representations, PReNet is compared with LDM that removes the hidden layers of PReNet (*i.e.*, the ψ function) and learns a Linear Direct Mapping (LDM) from the original data space to anomaly scores. The results show that eliminating the feature layer leads to over 10% loss in the average AUC-PR performance.
- **Deeper neural network**. We also explore the possibility of learning a deeper network in PReNet. A2H is a variant of PReNet, which deepens PReNet with additional two hidden (A2H) layers. Each added layer is regularized by an ℓ_2-norm regularizer and a dropout layer of 0.5 dropout rate. Although A2H performs well on some datasets like *rec* and *thyroid*, it fails on most of the other datasets. As a result, PReNet can still gain large average improvement over A2H. Thus, the default architecture used in PReNet is generally recommended.

Table 3: Unseen anomaly detection results. The models are trained with 'seen' anomaly classes to detect 'unseen' anomalies.

Anomaly Class		AUC-PR Results							AUC-ROC Results						
Seen	**Unseen**	**PReNet**	**DevNet**	**DSAD**	**FSNet**	**cFSNet**	**REPEN**	**iForest**	**PReNet**	**DevNet**	**DSAD**	**FSNet**	**cFSNet**	**REPEN**	**iForest**
dos	bac	**0.908**	**0.908**	0.890	0.772	0.841	0.590	0.050	**0.958**	0.956	0.924	0.932	0.943	0.889	0.741
dos, fuz	bac	**0.889**	0.886	0.879	0.555	0.679	0.438	0.050	**0.969**	0.967	0.918	0.921	0.941	0.886	0.741
fuz	bac	0.503	0.219	0.266	0.366	**0.524**	0.500	0.050	0.869	0.794	0.812	0.872	0.896	**0.899**	0.741
rec	bac	**0.752**	0.541	0.583	0.550	0.637	0.474	0.050	**0.965**	0.900	0.902	0.885	0.921	0.846	0.741
rec, dos	bac	0.834	**0.845**	0.833	0.48	0.537	0.548	0.050	0.977	**0.980**	0.978	0.924	0.942	0.882	0.741
rec, dos, fuz	bac	**0.706**	0.631	0.623	0.368	0.368	0.500	0.050	**0.971**	0.953	0.952	0.916	0.916	0.886	0.741
rec, fuz	bac	**0.711**	0.342	0.317	0.312	0.387	0.436	0.050	**0.969**	0.891	0.89	0.865	0.902	0.874	0.741
bac	dos	0.938	0.943	**0.961**	0.93	0.944	0.769	0.266	0.906	0.915	**0.945**	0.926	0.933	0.881	0.762
bac, fuz	dos	**0.932**	0.761	0.772	0.714	0.803	0.801	0.266	**0.958**	0.889	0.887	0.872	0.911	0.924	0.762
fuz	dos	0.811	0.644	0.68	0.774	0.803	**0.846**	0.266	0.855	0.792	0.805	0.826	0.846	**0.921**	0.762
rec	dos	**0.928**	0.846	0.855	0.798	0.831	0.771	0.266	**0.938**	0.883	0.887	0.825	0.846	0.875	0.762
rec, bac	dos	**0.891**	0.870	0.871	0.686	0.762	0.742	0.266	**0.944**	0.932	0.931	0.872	0.887	0.904	0.762
rec, bac, fuz	dos	**0.835**	0.610	0.699	0.572	0.627	0.641	0.266	**0.940**	0.861	0.886	0.821	0.858	0.890	0.762
rec, fuz	dos	**0.883**	0.718	0.673	0.670	0.764	0.749	0.266	**0.939**	0.874	0.868	0.834	0.88	0.903	0.762
bac	fuz	0.418	**0.420**	0.250	0.374	0.383	0.251	0.039	**0.752**	0.743	0.364	0.697	0.734	0.695	0.548
dos	fuz	0.418	0.427	0.325	0.288	0.324	**0.476**	0.039	0.708	0.737	0.508	0.606	0.708	**0.783**	0.548
dos, bac	fuz	0.375	0.371	0.322	0.273	0.301	**0.463**	0.039	0.842	0.833	0.646	0.837	**0.860**	0.828	0.548
rec	fuz	**0.462**	0.418	0.419	0.410	0.427	0.408	0.039	**0.878**	0.872	0.872	0.843	0.838	0.777	0.548
rec, bac	fuz	0.315	0.311	0.314	0.260	0.255	**0.319**	0.039	0.879	0.879	**0.880**	0.838	0.824	0.797	0.548
rec, bac, dos	fuz	0.294	0.246	0.249	0.206	0.189	**0.349**	0.039	**0.885**	0.878	0.878	0.846	0.853	0.832	0.548
rec, dos	fuz	0.349	0.375	0.366	0.276	0.306	**0.434**	0.039	0.850	**0.889**	0.871	0.822	0.837	0.800	0.548
bac	rec	**0.892**	0.890	0.576	0.689	0.718	0.592	0.132	**0.928**	0.926	0.489	0.693	0.713	0.741	0.534
bac, fuz	rec	**0.876**	0.804	0.831	0.672	0.692	0.554	0.132	**0.958**	0.943	0.947	0.879	0.897	0.822	0.534
dos	rec	**0.849**	0.846	0.739	0.615	0.718	0.686	0.132	**0.867**	0.865	0.677	0.636	0.727	0.778	0.534
dos, bac	rec	**0.768**	0.732	0.670	0.618	0.644	0.597	0.132	**0.908**	0.891	0.724	0.859	0.868	0.805	0.534
dos, bac, fuz	rec	**0.719**	0.716	0.629	0.540	0.586	0.447	0.132	**0.907**	**0.907**	0.820	0.871	0.892	0.783	0.534
dos, fuz	rec	**0.788**	0.772	0.615	0.631	0.644	0.542	0.132	**0.899**	0.885	0.663	0.842	0.863	0.800	0.534
fuz	rec	**0.797**	0.718	0.727	0.760	0.785	0.672	0.132	**0.890**	0.855	0.863	0.871	0.885	0.812	0.534
	Average	**0.709**	0.636	0.605	0.542	0.589	0.557	-	**0.904**	0.882	0.814	0.837	0.861	0.840	-
	P-value	-	0.001	0.000	0.000	0.000	0.000	-	-	0.001	0.000	0.000	0.000	0.000	-

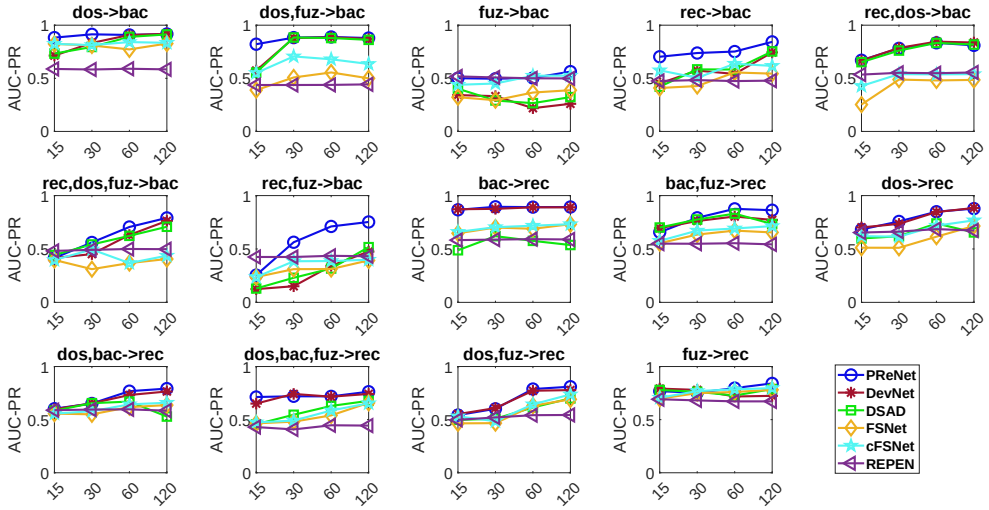

Figure 3: AUC-PR w.r.t. # of labeled anomalies. 'A -> B' means the models trained with attacks 'A' to detect unseen attacks 'B'.

- **Non-linear relation learning**. We compare PReNet with a variant of Non-linear Pairwise Relation (NPR) learning that adds a non-linear layer in-between the the concatenated features and the anomaly scoring layer. Similar to A2H, NRP can work better than PReNet on a few cases, but it is often too complex and has an overfitting problem on most datasets.

Sensitivity test. We evaluate the sensitivity of PReNet w.r.t. three key hyperparameters: $y_{\{\mathbf{x}_i,\mathbf{x}_j\}}$ and λ in Eqn. (5), and E in Eqn. (6).

- **Sensitivity w.r.t. pairwise class labels** $y_{\{\mathbf{x}_i,\mathbf{x}_j\}}$. This section examines the sensitivity of PReNet w.r.t. the synthetic ordinal pairwise relation class labels. We fix the ordinal label

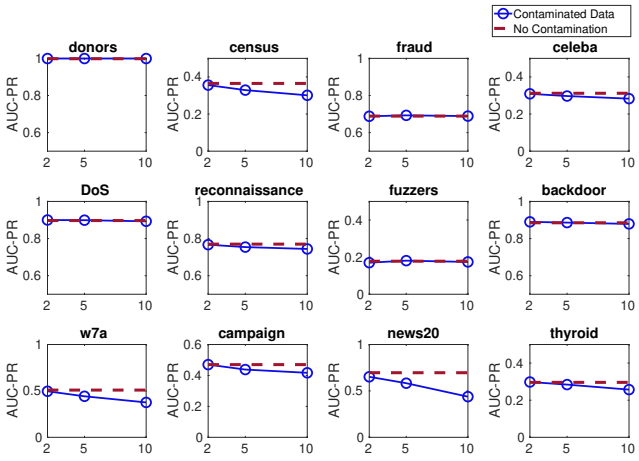

Figure 4: AUC-PR w.r.t. different contamination rates (%).

Table 4: AUC-PR results of ablation study.

Data	PReNet	TRM	iPReNet	LDM	A2H	NPR
donors	**1.000**	0.995	0.999	0.974	**1.000**	**1.000**
census	**0.356**	0.324	0.325	0.355	0.283	0.310
fraud	0.689	0.692	0.701	0.662	**0.708**	0.683
celeba	**0.309**	0.308	0.299	0.294	0.299	0.286
dos	0.900	0.887	0.887	0.839	0.877	**0.924**
rec	0.767	0.751	0.75	0.647	**0.877**	0.814
fuz	0.170	0.147	0.151	0.163	**0.178**	0.154
bac	0.890	0.879	0.879	0.805	0.863	**0.898**
w7a	**0.496**	0.467	0.482	0.406	0.415	0.393
campaign	**0.470**	0.423	0.402	0.406	0.246	0.370
news20	**0.652**	0.481	0.625	0.552	0.618	0.607
thyroid	0.298	0.269	0.248	0.201	**0.411**	0.341
P-value	-	0.002	0.003	0.001	0.413	0.328

for $y_{\{\mathbf{u},\mathbf{u}\}}$ to be zero, *i.e.*, $c_3 = 0$, and the same margin is set between $y_{\{\mathbf{u},\mathbf{u}\}}$ and $y_{\{\mathbf{a},\mathbf{u}\}}$ pairs, and between $y_{\{\mathbf{a},\mathbf{u}\}}$ and $y_{\{\mathbf{a},\mathbf{a}\}}$ pairs, *i.e.*, $(c_2 - c_3) = (c_1 - c_2) = m$. We test the sensitivity w.r.t. different values of the margin m. The AUC-PR results are shown in Figure 5. It is clear that PReNet is generally robust to different margin values. PReNet performs well even when setting a rather small margin, *e.g.*, $m = 0.25$. Larger margins are generally more desired, especially in some challenging datasets such as *thyroid* and *dos*.

- **Sensitivity w.r.t. ensemble size** E. This section investigates the sensitivity of PReNet w.r.t. the ensemble size E in Eqn. (6). The AUC-PR results are shown in Figure 5. PReNet performs very stably across all the 12 datasets. PReNet using small E performs similarly well as that using a large E, indicating that highly discriminative features are learned in PReNet. Increasing E may offer better detection accuracy on some datasets, but the improvement is often marginal.
- **Sensitivity w.r.t. regularization parameter** λ. This part investigates the sensitivity of PReNet w.r.t. a wide range of λ settings, $\lambda = \{0.001, 0.005, 0.01, 0.05, 0.1\}$, in Eqn. (5). The AUC-PR results are shown in Figure 5. PReNet is generally robust w.r.t. different λ values on all the 12 datasets, especially when λ is chosen in the range $[0.001, 0.01]$. When increasing λ to larger values such as 0.05 or 0.1, the AUC-PR of PReNet decreases on a few datasets, *e.g.*, *rec*, *bac*, *news20* and *thyroid*. This may be due to that given the limited number of labeled anomaly data, enforcing strong model regularization in PReNet can lead to underfitting on those datasets. Therefore, a small λ, *e.g.*, $\lambda = 0.01$, is generally recommended.

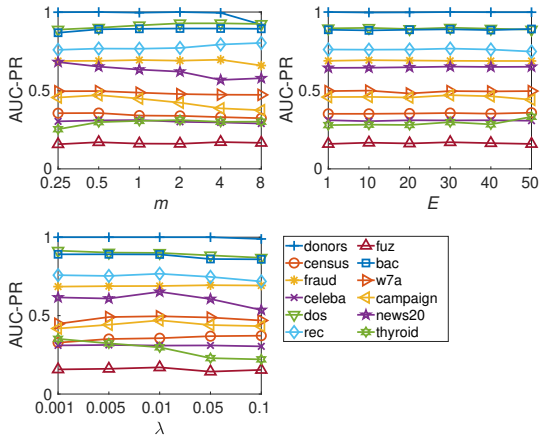

Figure 5: AUC-PR of PReNet w.r.t. three hyperparameters.

6 CONCLUSIONS

This paper explores the generalized semi-supervised anomaly detection problem and introduces a novel deep approach and its instantiated model PReNet to address the problem. The approach learns pairwise relation features and anomaly scores in a unified framework by three-way pairwise relation modeling. In doing so, it learns diverse abnormality and normality representations, alleviating the overfitting of the seen abnormalities. This is justified by the substantial improvement of PReNet over its variants and nine state-of-the-art competing methods that focus on learning more homogeneous normal/abnormal representations for detecting seen/unseen anomalies on 12 real-world datasets.

Particularly, our significantly improved precision-recall performance in unseen anomaly detection, *i.e.*, 10%-30%, is encouraging in that it is already very challenging to improve this metric for seen anomalies, and the challenge is further largely increased for the unseen anomalies. Our results also suggest that the labeled anomaly data, regardless of its scale and coverage of the anomaly classes, can be well leveraged to enable accurate anomaly detection in the wild. In future work, we plan to extend our approach to explore the supervision information from different domains of data for anomaly detection.

ACKNOWLEDGMENTS

C. Shen's participation was supported by National Key R&D Program of China (No. 2022ZD0118700). We thank Hezhe Qiao for helping obtain the results of XGBOD, PUMAD, and FEAWAD.

REFERENCES

[1] Charu C Aggarwal. 2017. *Outlier analysis*. Springer.

[2] Jessa Bekker and Jesse Davis. 2020. Learning from positive and unlabeled data: A survey. *Machine Learning* 109, 4 (2020), 719–760.

[3] Kendrick Boyd, Kevin H Eng, and C David Page. 2013. Area under the precision-recall curve: point estimates and confidence intervals. In *ECML/PKDD*. Springer, 451–466.

[4] Varun Chandola, Arindam Banerjee, and Vipin Kumar. 2009. Anomaly detection: A survey. *Comput. Surveys* 41, 3 (2009), 15.

[5] Dongyue Chen, Xinyue Tantai, Xingya Chang, Miaoting Tian, and Tong Jia. 2022. Weakly Supervised Anomaly Detection Based on Two-Step Cyclic Iterative PU Learning Strategy. *Neural Processing Letters* 54, 5 (2022), 4409–4426.

[6] Jinghui Chen, Saket Sathe, Charu Aggarwal, and Deepak Turaga. 2017. Outlier detection with autoencoder ensembles. In *SDM*. SIAM, 90–98.

[7] Yuanhong Chen, Yu Tian, Guansong Pang, and Gustavo Carneiro. 2022. Deep one-class classification via interpolated Gaussian descriptor. In *AAAI*, Vol. 36. 383–392.

[8] Choubo Ding, Guansong Pang, and Chunhua Shen. 2022. Catching both gray and black swans: Open-set supervised anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7388–7398.

[9] Kaize Ding, Qinghai Zhou, Hanghang Tong, and Huan Liu. 2021. Few-shot Network Anomaly Detection via Cross-network Meta-learning. In *WebConf*. 2448–2456.

[10] Charles Elkan and Keith Noto. 2008. Learning classifiers from only positive and unlabeled data. In *KDD*. ACM, 213–220.

[11] Li Fei-Fei, Rob Fergus, and Pietro Perona. 2006. One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28, 4 (2006), 594–611.

[12] Dong Gong, Lingqiao Liu, Vuong Le, Budhaditya Saha, Moussa Reda Mansour, Svetha Venkatesh, and Anton van den Hengel. 2019. Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In *ICCV*. 1705–1714.

[13] Nico Görnitz, Marius Kloft, Konrad Rieck, and Ulf Brefeld. 2013. Toward supervised anomaly detection. *Journal of Artificial Intelligence Research* 46 (2013), 235–262.

[14] Sachin Goyal, Aditi Raghunathan, Moksh Jain, Harsha Vardhan Simhadri, and Prateek Jain. 2020. DROCC: Deep robust one-class classification. In *ICML*. PMLR, 3711–3721.

[15] Dino Ienco, Ruggero G Pensa, and Rosa Meo. 2017. A semisupervised approach to the detection and characterization of outliers in categorical data. *IEEE Transactions on Neural Networks and Learning Systems* 28, 5 (2017), 1017–1029.

[16] Minqi Jiang, Chaochuan Hou, Ao Zheng, Xiyang Hu, Songqiao Han, Hailiang Huang, Xiangnan He, Philip S Yu, and Yue Zhao. 2023. Weakly supervised anomaly detection: A survey. *arXiv preprint arXiv:2302.04549* (2023).

[17] Huidong Jin, Jie Chen, Hongxing He, Chris Kelman, Damien McAullay, and Christine M O'Keefe. 2010. Signaling potential adverse drug reactions from administrative health databases. *IEEE Transactions on Knowledge and Data Engineering* 22, 6 (2010), 839–853.

[18] Huidong Jin, Jie Chen, Hongxing He, Graham J Williams, Chris Kelman, and Christine M O'Keefe. 2008. Mining unexpected temporal associations: applications in detecting adverse drug reactions. *IEEE Transactions on Information Technology in Biomedicine* 12, 4 (2008), 488–500.

[19] Hyunjun Ju, Dongha Lee, Junyoung Hwang, Junghyun Namkung, and Hwanjo Yu. 2020. PUMAD: PU metric learning for anomaly detection. *Information Sciences* 523 (2020), 167–183.

[20] Hannah R Kerner, Kiri L Wagstaff, Brian D Bue, Danika F Wellington, Samantha Jacob, Paul Horton, James F Bell, Chiman Kwan, and Heni Ben Amor. 2020. Comparison of novelty detection methods for multispectral images in rover-based planetary exploration missions. *Data Mining and Knowledge Discovery* 34 (2020), 1642–1675.

[21] Viet Lai, Franck Dernoncourt, and Thien Huu Nguyen. 2021. Learning prototype representations across few-shot tasks for event detection. In *EMNLP*. 5270–5277.

[22] Dan Li, Dacheng Chen, Baihong Jin, Lei Shi, Jonathan Goh, and See-Kiong Ng. 2019. MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial networks. In *ICANN*. Springer, 703–716.

[23] Xiaoli Li and Bing Liu. 2003. Learning to classify texts using positive and unlabeled data. In *IJCAI*, Vol. 3. 587–592.

[24] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. 2012. Isolation-based anomaly detection. *ACM Transactions on Knowledge Discovery from Data* 6, 1 (2012), 3.

[25] Philipp Liznerski, Lukas Ruff, Robert A Vandermeulen, Billy Joe Franks, Marius Kloft, and Klaus-Robert Müller. 2021. Explainable deep one-class classification. In *ICLR*.

[26] Yuxuan Luo, Shaoyin Cheng, Chong Liu, and Fan Jiang. 2018. PU learning in payload-based web anomaly detection. In *2018 Third International Conference on Security of Smart Cities, Industrial Control System and Communications (SSIC)*. IEEE, 1–5.

[27] Mary McGlohon, Stephen Bay, Markus G Anderle, David M Steier, and Christos Faloutsos. 2009. SNARE: A link analytic system for graph labeling and risk detection. In *KDD*. ACM, 1265–1274.

[28] Nour Moustafa and Jill Slay. 2015. UNSW-NB15: a comprehensive data set for network intrusion detection systems. In *Military Communications and Information Systems Conference, 2015*. 1–6.

[29] Keith Noto, Carla Brodley, and Donna Slonim. 2012. FRaC: a feature-modeling approach for semi-supervised and unsupervised anomaly detection. *Data Mining and Knowledge Discovery* 25, 1 (2012), 109–133.

[30] Guansong Pang, Longbing Cao, Ling Chen, and Huan Liu. 2018. Learning Representations of Ultrahigh-dimensional Data for Random Distance-based Outlier Detection. In *KDD*. 2041–2050.

[31] Guansong Pang, Choubo Ding, Chunhua Shen, and Anton van den Hengel. 2021. Explainable Deep Few-shot Anomaly Detection with Deviation Networks. *arXiv preprint arXiv:2108.00462* (2021).

[32] Guansong Pang, Chunhua Shen, Longbing Cao, and Anton van den Hengel. 2021. Deep Learning for Anomaly Detection: A Review. *Comput. Surveys* 54, 2 (2021), 1–38.

[33] Guansong Pang, Chunhua Shen, and Anton van den Hengel. 2019. Deep Anomaly Detection with Deviation Networks. In *KDD*. ACM, 353–362.

[34] Guansong Pang, Anton van den Hengel, Chunhua Shen, and Longbing Cao. 2021. Toward deep supervised anomaly detection: Reinforcement learning from partially labeled anomaly data. In *KDD*. 1298–1308.

[35] Guansong Pang, Cheng Yan, Chunhua Shen, Anton van den Hengel, and Xiao Bai. 2020. Self-trained deep ordinal regression for end-to-end video anomaly detection. In *CVPR*. 12173–12182.

[36] Hyunjong Park, Jongyoun Noh, and Bumsub Ham. 2020. Learning memory-guided normality for anomaly detection. In *CVPR*. 14372–14381.

[37] Lorenzo Perini, Paul Buerkner, and Arto Klami. 2022. Estimating the Contamination Factor's Distribution in Unsupervised Anomaly Detection. *arXiv preprint arXiv:2210.10487* (2022).

[38] Lorenzo Perini, Vincent Vercruyssen, and Jesse Davis. 2020. Class prior estimation in active positive and unlabeled learning. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence and the 17th Pacific Rim International Conference on Artificial Intelligence (IJCAI-PRICAI 2020)*. IJCAI-PRICAI, 2915–2921.

[39] Lorenzo Perini, Vincent Vercruyssen, and Jesse Davis. 2022. Transferring the contamination factor between anomaly detection domains by shape similarity. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 4128–4136.

[40] Tal Reiss, Niv Cohen, Liron Bergman, and Yedid Hoshen. 2021. Panda: Adapting pretrained features for anomaly detection and segmentation. In *CVPR*. 2806–2814.

[41] Karsten Roth, Latha Pemula, Joaquin Zepeda, Bernhard Schölkopf, Thomas Brox, and Peter Gehler. 2022. Towards total recall in industrial anomaly detection. In *CVPR*. 14318–14328.

[42] Lukas Ruff, Nico Görnitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Robert Vandermeulen, Alexander Binder, Emmanuel Müller, and Marius Kloft. 2018. Deep one-class classification. In *ICML*. 4390–4399.

[43] Lukas Ruff, Robert A Vandermeulen, Nico Görnitz, Alexander Binder, Emmanuel Müller, Klaus-Robert Müller, and Marius Kloft. 2020. Deep Semi-Supervised Anomaly Detection. In *ICLR*.

[44] Emanuele Sansone, Francesco GB De Natale, and Zhi-Hua Zhou. 2018. Efficient training for positive unlabeled learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2018).

[45] Thomas Schlegl, Philipp Seeböck, Sebastian M Waldstein, Ursula Schmidt-Erfurth, and Georg Langs. 2017. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In *IPMI*. Springer, Cham, 146–157.

[46] Yiyuan She and Art B Owen. 2011. Outlier detection using nonconvex penalized regression. *J. Amer. Statist. Assoc.* 106, 494 (2011), 626–639.

[47] Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *NeurIPS*. 4077–4087.

[48] Kihyuk Sohn, Chun-Liang Li, Jinsung Yoon, Minho Jin, and Tomas Pfister. 2021. Learning and evaluating representations for deep one-class classification. In *ICLR*.

[49] Hanyu Song, Peizhao Li, and Hongfu Liu. 2021. Deep Clustering based Fair Outlier Detection. In *KDD*. 1481–1489.

[50] Philip Sperl, Jan-Philipp Schulze, and Konstantin Böttinger. 2020. Activation anomaly analysis. In *ECML/PKDD*. Springer, 69–84.

[51] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. 2018. Learning to compare: Relation network for few-shot learning. In *CVPR*. 1199–1208.

[52] Acar Tamersoy, Kevin Roundy, and Duen Horng Chau. 2014. Guilt by association: Large scale malware detection by mining file-relation graphs. In *KDD*. 1524–1533.

[53] Vincent Vercruyssen, Wannes Meert, Gust Verbruggen, Koen Maes, Ruben Baumer, and Jesse Davis. 2018. Semi-supervised anomaly detection with an application to water analytics. In *ICDM*, Vol. 2018. IEEE, 527–536.

[54] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. 2016. Matching networks for one shot learning. In *NIPS*. 3630–3638.

[55] Hu Wang, Guansong Pang, Chunhua Shen, and Congbo Ma. 2020. Unsupervised representation learning by predicting random distances. *IJCAI* (2020).

[56] Lingyu Wang, Sushil Jajodia, Anoop Singhal, Pengsu Cheng, and Steven Noel. 2013. k-zero day safety: A network security metric for measuring the risk of unknown vulnerabilities. *IEEE Transactions on Dependable and Secure Computing* 11, 1 (2013), 30–44.

[57] Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. 2020. Generalizing from a few examples: A survey on few-shot learning. *Comput. Surveys* 53, 3 (2020), 1–34.

[58] RF Woolson. 2007. Wilcoxon signed-rank test. *Wiley Encyclopedia of Clinical Trials* (2007), 1–3.

[59] Chun Yu and Weixin Yao. 2017. Robust linear regression: A review and comparison. *Communications in Statistics-Simulation and Computation* 46, 8 (2017), 6261–6282.

[60] Houssam Zenati, Manon Romain, Chuan-Sheng Foo, Bruno Lecouat, and Vijay Chandrasekhar. 2018. Adversarially Learned Anomaly Detection. In *ICDM*. IEEE, 727–736.

[61] Chuxu Zhang, Dongjin Song, Yuncong Chen, Xinyang Feng, Cristian Lumezanu, Wei Cheng, Jingchao Ni, Bo Zong, Haifeng Chen, and Nitesh V Chawla. 2019. A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data. In *AAAI*, Vol. 33. 1409–1416.

[62] Ke Zhang, Marcus Hutter, and Huidong Jin. 2009. A new local distance-based outlier detection approach for scattered real-world data. In *PAKDD*. 813–822.

[63] Xianchao Zhang, Jie Mu, Xiaotong Zhang, Han Liu, Linlin Zong, and Yuangang Li. 2022. Deep anomaly detection with self-supervised learning and adversarial training. *Pattern Recognition* 121 (2022), 108234.

[64] Ya-Lin Zhang, Longfei Li, Jun Zhou, Xiaolong Li, and Zhi-Hua Zhou. 2018. Anomaly detection with partially observed anomalies. In *WWW Companion*. 639–646.

[65] Yue Zhao and Maciej K Hryniewicki. 2018. Xgbod: improving supervised outlier detection with unsupervised representation learning. In *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–8.

[66] Panpan Zheng, Shuhan Yuan, Xintao Wu, Jun Li, and Aidong Lu. 2019. One-class adversarial nets for fraud detection. In *AAAI*, Vol. 33. 1286–1293.

[67] Shuang Zhou, Xiao Huang, Ninghao Liu, Qiaoyu Tan, and Fu-Lai Chung. 2022. Unseen Anomaly Detection on Networks via Multi-Hypersphere Learning. In *SDM*. SIAM, 262–270.

[68] Yingjie Zhou, Xucheng Song, Yanru Zhang, Fanxing Liu, Ce Zhu, and Lingqiao Liu. 2022. Feature encoding with autoencoders for weakly supervised anomaly detection. *IEEE Transactions on Neural Networks and Learning Systems* 33, 6 (2022), 2454–2465.

[69] Arthur Zimek, Matthew Gaudet, Ricardo JGB Campello, and Jörg Sander. 2013. Subsampling for efficient and effective unsupervised outlier detection ensembles. In *KDD*. ACM, 428–436.

[70] Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. 2018. Deep autoencoding Gaussian mixture model for unsupervised anomaly detection. In *ICLR*.

A DETAILED EXPERIMENT SETUP

A.1 Datasets

A.1.1 Seen Anomaly Detection Datasets. The 12 seen anomaly detection datasets can be accessed via the links in Table 5. More specifically, the *donors* data is taken from KDD Cup 2014 for predicting the excitement of donation projects, with exceptionally exciting projects used as anomalies (6.0% of all data instances). The *census* data is extracted from the US census bureau database, in which we aim to detect the rare high-income persons (6.0%). *fraud* is for fraudulent credit card transaction detection, with fraudulent transactions (0.2%) as anomalies. *celeba* contains more than 200K celebrity images, each with 40 attribute annotations. We use the bald attribute as our detection target, in which the scarce bald celebrities (3.0%) are treated as anomalies and the other 39 attributes form the feature space. The *dos*, *rec*, *fuz* and *bac* datasets are derived from a popular intrusion detection dataset called *UNSW-NB15* [28] with the respective *DoS* (15.0%), *reconnaissance* (13.1%), *fuzzers* (3.1%) and *backdoor* (2.4%) attacks as anomalies against the 'normal' class. *w7a* is a web page classification dataset, with the minority classes (3.0%) as anomalies. *campaign* is a dataset of bank marketing campaigns, with rare positive campaigning records (11.3%) as anomalies. *news20* is one of the most popular text classification corpora, which is converted into anomaly detection data via random downsampling of the minority class (5.0%) based on [24, 69]. *thyroid* is for disease detection, in which the anomalies are the hypothyroid patients (7.4%). Seven of these datasets contain real anomalies,including *donors*, *fraud*, *dos*, *rec*, *fuz*, *bac* and *thyroid*. The other five datasets contain semantically real anomalies, *i.e.*, they are rare and very different from the majority of data instances. So, they serve as a good testbed for the evaluation of anomaly detection techniques.

Table 5: Links for accessing the datasets

Data	Link
donors	https://www.kaggle.com/c/kdd-cup-2014-predicting-excitement-at-donors-choose
census	https://archive.ics.uci.edu/ml/datasets/census+income
fraud	https://www.kaggle.com/c/1056lab-credit-card-fraud-detection
celeba	http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html
dos	https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/
rec	https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/
fuz	https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/
bac	https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/
w7a	https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/
campaign	https://archive.ics.uci.edu/ml/datasets/bank+marketing
news20	https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/
thyroid	https://www.openml.org/d/40497

To replicate the real-world scenarios where we have a few labeled anomalies and large unlabeled data, we first have a stratified split of the anomalies and normal instances into two subsets, with 80% data as training data and the other 20% data as a holdup test set. Since the unlabeled data is often anomaly-contaminated, we then combine some randomly selected anomalies with the whole normal training data instances to form the unlabeled dataset $\mathcal{U}$. We further randomly sample a limited number of anomalies from the anomaly class to form the labeled anomaly set $\mathcal{A}$. The resulting sample size and dimensionality of the datasets are shown in Table 2.

A.1.2 Unseen Anomaly Detection Datasets. Table 3 presents the 28 datasets for the evaluation of detecting unseen anomalies. These datasets are derived from the above four intrusion attack datasets *dos*, *rec*, *fuz* and *bac*, with data instances spanned by the same feature space. To guarantee that the evaluation data contains unseen anomalies, the anomaly class in one of these four datasets is held up for evaluation, while the anomalies in any combinations of the remaining three datasets are combined to form the pool of seen anomalies. The type of the holdup anomalies is always different from that in the anomaly pool and can be safely treated as unseen anomalies. We have 28 possible permutations under this setting, resulting in 28 datasets with different seen and/or unseen anomalies. For the training, $\mathcal{A}$ contains the anomalies sampled from the seen anomalies pool, while the evaluation data is composed of the holdup unseen anomaly class and the 20% holdup normal instances. Note that the other eight datasets in Table 2 cannot be used in evaluating unseen anomaly detection as they contain only one anomaly class and they are from different data sources and feature spaces.

A.2 Implementation Details

A.2.1 Packages. PReNet is implemented using Tensorflow/Keras. The main packages and their versions used in this work are provided as follows:

- keras==2.3.1
- numpy==1.16.2
- pandas==0.23.4
- scikit-learn==0.20.0
- scipy==1.1.0
- tensorboard==1.14.0
- tensorflow==1.14.0

A.2.2 Hyperparameter Settings. Since our experiments focus on unordered multidimensional data, multilayer perceptron networks are used. Similar to [30, 33], we empirically found that all deep methods using an architecture with one hidden layer perform better and more stably than using two or more hidden layers. This may be due to the limit of the available labeled data. Following [30, 33], one hidden layer with 20 neural units is used in PReNet. The ReLu activation function $g(a) = max(0, a)$ is used. An ℓ_2-norm regularizer with the hyperparameter setting $\lambda = 0.01$ is applied to avoid overfitting. The RMSprop optimizer with the learning rate 0.001 is used. The same network architecture is used in the competing methods DevNet [33], REPEN [30], Deep SAD (DSAD) [43], FSNet [47] and its variant cFSNet. iForest with the recommended settings [24], 100 isolation trees and 256 subsampling size, are used in our experiments.

All deep detectors are trained using 50 epochs, with 20 batches per epoch. The batch size is probed in {8, 16, 32, 64, 128, 256, 512}. The best fits, 512 in PReNet, DevNet and DSAD, 256 in FSNet and REPEN, are used by default. cFSNet uses the same settings as FSNet. Similar to PReNet, oversampling is also applied to the labeled anomaly set $\mathcal{A}$ to well train the deep detection models of DevNet, REPEN, DSAD, FSNet and cFSNet.

B ADDITIONAL EMPIRICAL RESULTS

B.1 Comparison to Three More Methods

We also compare PReNet to three more methods XGBOD [65], PUMAD [19], and FEAWAD [68] on both seen and unseen anomaly detection datasets, with the results shown in Tables 6 and 7 respectively. The official implementation of XGBOD and FEAWAD was used to perform the experiments. The code of PUMAD is not released; we use our own implementation based on a metric learning similar to REPEN. As for XGBOD, we used one-class SVM and iForest to produce new features only; *k*NN and LOF were excluded due to prohibitive computational cost on large datasets. All the other settings in XGBOD remain unchanged.

Table 6: AUC-PR results for seen AD datasets. OOM denotes an out-of-memory issue in a GeForce RTX 3090 24GB GPU.

Dataset	PReNet	XGBOD	PUMAD	FEAWAD
donors	**1.000**	0.178	0.215	**1.000**
census	**0.356**	0.061	0.129	0.252
fraud	**0.689**	0.408	0.423	0.670
celeba	**0.309**	0.081	0.143	0.225
dos	**0.900**	0.429	0.328	0.827
rec	0.767	0.041	0.379	**0.852**
fuz	**0.170**	0.097	0.065	0.118
bac	**0.890**	0.145	0.120	0.811
w7a	**0.496**	0.224	0.072	0.406
campaign	**0.470**	0.323	0.302	0.365
news20	**0.652**	0.076	0.107	OOM
thyroid	0.298	0.262	0.163	**0.383**

The results show that these three methods, especially FEAWAD, can work well on some datasets, but they still substantially underperform our method PReNet on most datasets.

B.2 Performance Ranking of All Methods

To have a holistic comparison of all 10 detectors, we calculate the average (ordinal and percentile) rank of each method based on its detection performance in both seen and unseen AD settings. The results are shown in Table 8, where an average ordinal rank of one (or a percentile of 1.00) is the perfect performance, indicating the method is always the best performer compared to all other methods across all datasets. That is, lower rank (or higher percentile) indicates better performance.

The results show that PReNet is the best performer in both settings, outperforming all nine competing methods for 94.2% and 90.2% of cases on 12 seen anomaly detection datasets and 28 unseen anomaly detection datasets respectively. It is followed by DevNet and DSAD in seen AD, and DevNet and FEAWAD in unseen AD. When using a Conover post-hoc test at the 95% confidence level, PReNet performs significant better than all methods except DevNet and DSAD in seen AD; it significantly outperforms all other nine methods in unseen AD.

Table 7: AUC-PR results for unseen AD datasets.

Seen	Unseen	PReNet	XGBOD	PUMAD	FEAWAD
dos	bac	**0.908**	0.429	0.360	0.832
dos, fuz	bac	**0.889**	0.372	0.274	0.703
fuz	bac	**0.503**	0.145	0.061	0.131
rec	bac	0.752	0.224	0.497	**0.851**
rec, dos	bac	**0.834**	0.367	0.321	0.831
rec, dos, fuz	bac	**0.706**	0.378	0.143	0.610
rec, fuz	bac	**0.711**	0.284	0.138	0.533
bac	dos	**0.938**	0.097	0.114	0.785
bac, fuz	dos	**0.932**	0.284	0.053	0.176
fuz	dos	**0.811**	0.145	0.061	0.128
rec	dos	**0.928**	0.224	0.388	0.848
rec, bac	dos	**0.891**	0.244	0.164	0.772
rec, bac, fuz	dos	**0.835**	0.178	0.199	0.550
rec, fuz	dos	**0.883**	0.228	0.136	0.556
bac	fuz	0.418	0.097	0.114	**0.769**
dos	fuz	0.418	0.429	0.328	**0.842**
dos, bac	fuz	0.375	0.284	0.189	**0.805**
rec	fuz	0.462	0.224	0.388	**0.860**
rec, bac	fuz	0.315	0.244	0.166	**0.790**
rec, bac, dos	fuz	0.294	0.342	0.241	**0.734**
rec, dos	fuz	0.349	0.361	0.321	**0.840**
bac	rec	**0.892**	0.097	0.155	0.684
bac, fuz	rec	**0.876**	0.054	0.042	0.192
dos	rec	**0.849**	0.429	0.335	0.818
dos, bac	rec	0.768	0.284	0.163	**0.775**
dos, bac, fuz	rec	**0.719**	0.163	0.153	0.700
dos, fuz	rec	**0.788**	0.371	0.280	0.700
fuz	rec	**0.797**	0.145	0.071	0.745

Table 8: Average (ordinal and percentile) rank of methods based on AUC-PR for seen and unseen AD across the 12 and 28 datasets, respectively. The methods are sequentially sorted based on the ordinal rank of seen and unseen AD.

	Seen AD		Unseen AD	
Method	Ordinal	Percentile	Ordinal	Percentile
PReNet	**1.583**	**0.942**	**1.982**	**0.902**
DevNet	2.667	0.832	3.661	0.734
DSAD	3.292	0.769	4.143	0.686
FEAWAD	3.955	0.705	3.786	0.721
cFSNet	5.792	0.517	4.518	0.648
REPEN	6.583	0.439	4.893	0.611
FSNet	6.625	0.432	6.054	0.495
XGBOD	7.417	0.352	7.857	0.314
PUMAD	7.583	0.337	8.750	0.225
iForest	9.000	0.193	9.357	0.164