Weakly-supervised Anomaly Detection with Adaptive Score Distribution Discrimination

Minqi Jiang
Shanghai University of Finance and Economics
ShangHai
jiangmq95@163.com

Songqiao Han
Shanghai University of Finance and Economics
ShangHai
han.songqiao@shufe.edu.cn

Hailiang Huang†
Shanghai University of Finance and Economics
ShangHai
hlhuang@shufe.edu.cn

ABSTRACT

Recent studies give more attention to weakly-supervised anomaly detection (AD), which can leverage a handful of labeled anomalies along with abundant unlabeled data. Existing methods usually rely on manually predefined score target(s), e.g., prior constant or margin hyperparameter(s), to realize discrimination in anomaly scores between normal and abnormal data. However, such methods would be vulnerable to the existence of anomaly contamination in the unlabeled data, and also lack adaptation to different data scenarios.

In this paper, we propose to optimize the anomaly scoring function from the view of score distribution, thus better retaining the diversity and more fine-grained information of input data, especially of the unlabeled data that contains anomaly noises in the weakly-supervised AD problem. We design a novel loss function called Overlap Loss that minimizes the overlap area between the score distributions of normal and abnormal samples, which no longer depends on prior anomaly score targets and thus acquires adaptability to various datasets. Overlap loss consists of *Score Distribution Estimator* and *Overlap Area Calculation*, which are introduced to overcome challenges when estimating arbitrary score distributions, and to ensure the boundness of training loss. As a general loss component, Overlap loss can be effectively integrated into multiple network architectures for constructing AD models. Extensive experimental results indicate that Overlap loss based AD models significantly outperform their state-of-the-art counterparts, and achieve better performance for different types of anomalies.

PVLDB Reference Format:
Minqi Jiang, Songqiao Han, and Hailiang Huang†. Weakly-supervised Anomaly Detection with Adaptive Score Distribution Discrimination. PVLDB, 16(1): XXX-XXX, 2023. doi:XX.XX/XXX.XX

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/Minqi824/Overlap.

1 INTRODUCTION

Anomaly detection (AD) is the task of identifying unusual instances that deviate significantly from the majority of data, which has been applied in wide-ranging domains, such as social media analysis [43, 70, 75], rare disease detection [54, 76], intrusion detection [31], and financial fraud detection [24, 53].

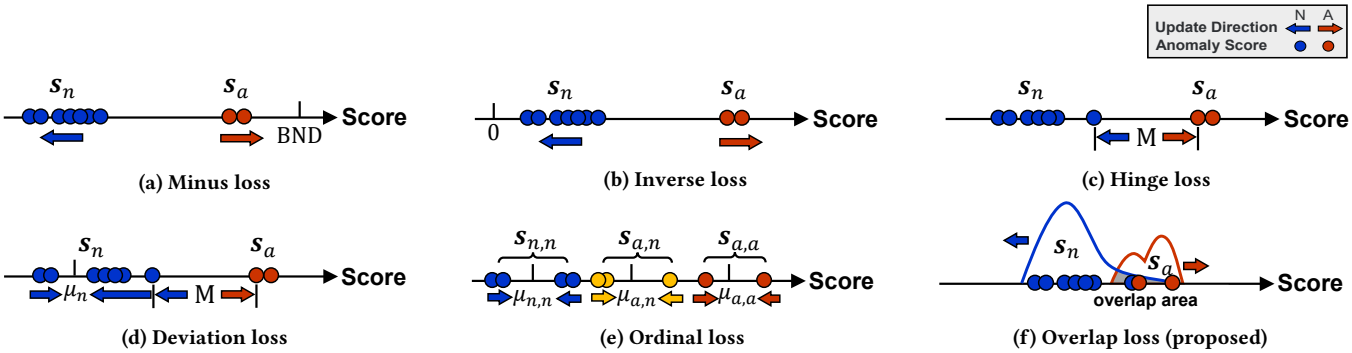
Previous research efforts [23, 36, 39, 41, 58, 60, 81] focus on unsupervised AD which does not require any labeled training data, but unsupervised methods lack any guidance of true anomalies [48, 49, 51]. Therefore recent studies propose to learn valuable distinguishing features from a few labeled anomalies that may be identified by domain experts in practice, which is termed as the weakly-supervised anomaly detection[1] [14, 25, 48, 50, 51, 79].

Current weakly-supervised methods mainly devise specific forms of loss functions to leverage such limited label information, including representation learning based Minus loss in Unlearning [14], Inverse loss in DeepSAD [59] and Hinge loss in REPEN [48], as we summarized in Figure 1a~1c. In these methods, an anomaly score is generated based on the learned feature transformation of input data, such as the reconstruction error or the embedding distance. However, optimization in the representation space would lead to data-inefficient learning and suboptimal anomaly scoring [49, 51]. Therefore several works [25, 50, 51, 79] fulfill an end-to-end learning fashion of anomaly score to obtain better performance, designing loss functions to map input instances to their corresponding anomaly score target(s), or to predefine a margin hyperparameter to realize the difference in anomaly scores between unlabeled samples and labeled anomalies, as shown in Figure 1d~1e. Nevertheless, such a predefined score target or margin would constrain the model's adaptability to different datasets, and further tuning these hyperparameters for realizing adaptation is often difficult, considering the scarcity of labeled data in the weakly-supervised AD scenarios.

Although most existing DNN-based AD methods realize a pointwise optimization [42] of anomaly scores, i.e., they calculate the pointwise loss differences between the estimated anomaly score and the training target, we revisit these loss functions and indicate that their learned anomaly scores often exhibit a form similar to score distribution and can be well categorized as (shown in Figure 1): (i) Enlarge the difference of the *means* of anomaly scores between normal samples and anomalies. This corresponds to the Minus loss, Inverse loss, and Hinge loss, where the anomaly scores of these two types of samples are gradually separated during the gradient update process. (ii) Reduce the *variances* of anomaly scores by predefining anomaly score targets. A case in point is the Ordinal loss, which maps the instance pairs to multiple predefined

Proceedings of the VLDB Endowment, Vol. 16, No. 1 ISSN 2150-8097.
doi:XX.XX/XXX.XX
†Corresponding authors.

[1]According to the definition in [80], this actually refers to the incomplete supervision problem in weakly-supervised learning, where besides a great number of unlabeled instances, there exist a limited number of labeled anomalies. We use the *unlabeled samples* and *normal samples* interchangeably in this paper without causing ambiguity.



(a) Minus loss (b) Inverse loss (c) Hinge loss

(d) Deviation loss (e) Ordinal loss (f) Overlap loss (proposed)

Figure 1: Loss function comparison. (a) Minus loss [14] provides opposite update directions of anomaly scores between the normal (blue) and abnormal (red) data. A predetermined upper bound BND is applied to prevent the exploding loss issue [14]. (b) Inverse loss [59] is similar to minus loss, but bounded by itself. (c) Hinge loss [48] uses a predefined hyperparameter M to ensure a margin of at least M in anomaly scores. (d) Deviation loss [25, 51, 79] enforces the anomaly scores of normal data to follow a standard Gaussian distribution, and pushes that of anomalies at least M margin. (e) Ordinal loss [50] preassigns three constants $\mu_{n,n}$, $\mu_{a,n}$ and $\mu_{a,a}$ as the training targets to keep score margins among three types of instance pairs. (f) Proposed Overlap loss estimates score distributions of the normal and abnormal data and further minimizes their overlap area without requiring predefined anomaly score target(s).

constants. (iii) Combine the above two methods, e.g. the Deviation loss, which reduces the *variance* of the normal scores by mapping them to the predefined value μ_n and enlarges the score difference in the *means* of anomaly scores by the M margin hyperparameters.

We aim to acquire an anomaly scoring function capable of generating adaptive score distribution for various data scenarios, thus unifying the loss categories discussed above. This idea is also inspired by the work [12, 56] in the computer vision (CV) domain, where they distinguish different classes by penalizing the class distribution overlap in the multi-classification tasks. Compared to their work, we focus on studying distribution overlap in the anomaly scoring space instead of the representation space, since both the curse-of-dimensionality problem and the scarcity of labeled data would make the embedding more sparse in the representation space, which brings more difficulties for estimating distribution overlap. To the best of our knowledge, anomaly detection based on the score distribution overlap has not been previously studied.

To address the above issues, we devise the Overlap loss that minimizes score distribution overlap between normal samples and anomalies, therefore depending on the model itself to decide the suitable means and variances of anomaly score distributions. This kind of adaptability eliminates the dependency on the predefined anomaly score targets. However, a non-trivial challenge in the weakly-supervised AD is to estimate arbitrary distributions of anomaly scores, which are caused by both the scarcity of labeled anomalies and anomaly noises in the unlabeled data. In the Overlap loss, we design a simple and effective method to estimate the overlap area of arbitrary score distributions, while ensuring a correct order in the output anomaly scores and the boundness of training loss to better achieve stability in model training.

The main contributions of this paper can be summarized as follows: **(1)** We propose the Overlap loss for weakly-supervised AD, which achieves adaptive score distribution discrimination between normal and abnormal data, realizing sufficient global insight of anomaly scores in an end-to-end gradient update fashion. **(2)** We verify the effectiveness of the proposed Overlap loss on several network architectures covering both anomaly detection and classification tasks. Extensive experimental results on 25 datasets suggest that the proposed Overlap loss could be served as a basis for further development in AD tasks. We open-source the proposed method, related codes, and all testing datasets for AD communities at https://github.com/Minqi824/Overlap. **(3)** We decouple the loss functions from several popular AD models and analyze them in a unified framework, including embedding variation and network parameter changes. Moreover, we investigate the detection performance of different loss functions on various types of anomalies, therefore further exploring the pros and cons of these methods.

2 RELATED WORK

A desirable anomaly detection approach should produce not only a binary output (normal or abnormal) but also assign a degree of being an anomaly (anomaly score) to each observation [74]. Prior literature can be divided into two categories, i.e., AD algorithms without or with supervision. The former assumes that no labeled data is available during the model training stage and is proposed with different assumptions of data distribution [2], e.g., anomalies located in low-density regions, and their performance often depends on the agreement between the input data and the algorithm assumption(s). The latter assumes that although acquiring labels is often time-consuming and effort-intensive, one may have access to a limited number of labeled samples, which could be verified by some domain experts or automatic detecting systems.

AD Algorithms without Supervision. Typical anomaly detection methods are constructed for learning anomaly patterns in an unsupervised manner. These include shallow unsupervised models like CBLOF [23] and ECOD [36], or ensemble method Isolation Forest [39]. More recently, deep learning (DL) techniques like DeepSVDD [58] and GAN-based MO-GAAL [41] have been proposed for improving the performance of unsupervised AD tasks.

AD Algorithms with Supervision. Unsupervised methods can not achieve satisfactory performance in practical applications without the guidance of labeled data. Therefore several studies have also investigated utilizing partially labeled data to improve detection performance, which can be summarized into the following three categories:

(i) Semi-supervised AD methods that are trained only on labeled normal samples, and detect anomalies that deviate from the normal representation learned in the training process [3, 4, 71, 72].

(ii) Semi- and weakly-supervised AD methods that additionally leverage a limited number of labeled anomalies. A common problem of the above methods leveraging only normal samples is that many of the anomalies they identify are data noises or uninteresting data instances due to the lack of prior knowledge about the abnormal behaviors [48, 49, 51]. This results in the development of semi- and weakly-supervised AD methods, which not only learn from numerous unlabeled data but also utilize limited information of labeled anomalies.

Among them, the Unlearning [14] method uses the minus loss form to provide an opposite direction of gradient update between the reconstruction error of the normal data and anomalies. DeepSAD [59] employs the inverse loss to penalize the inverse of the embedding distance such that the representation of anomalies must be mapped further away from the initial center of the hypersphere. REPEN [48] introduces a ranking model-based framework, which applies the hinge loss to encourage a distance separation of low-dimensional representation between normal samples and anomalies. However, the above methods perform an indirect optimization of the anomaly scoring function, i.e., they compute the anomaly score either based on the reconstruction error or the embedding distance. Such indirect optimization of anomaly score would lead to data-inefficient learning and suboptimal detection performance [49, 51].

Therefore, several works are proposed to realize end-to-end learning of anomaly score. Specified with the deviation loss, DevNet [51] leverages a prior probability and a margin hyperparameter to enforce significant deviations in anomaly scores between normal and abnormal data. FEAWAD [79] incorporates the DAGMM [81] network architecture with the deviation loss, for the better use of the information among hidden representation, reconstruction residual vector and reconstruction error transformed by the auto-encoder [7]. Similarly, SG-AE [25] leverages the autoencoder-based reconstruction error as a guide for modeling the score discrepancy, where the anomaly scores of both normal and abnormal data are optimized as in the DevNet. PReNet [50] formulates the scoring function as a pairwise relation learning task, where it defines three constant targets to enforce large margins among the anomaly scores of three types of instance pairs.

(iii) Fully-supervised methods are not specific for AD tasks in general [20]. Previous studies [6, 47] often use existing binary classifiers for this purpose such as Random Forest and MLP. One known risk of supervised methods is that ground truth labels maybe not necessarily accurate enough (i.e., there often exist some unlabeled anomaly noises in normal samples) to capture all types of anomalies, therefore these supervised methods may fail to detect unknown types of anomalies [22, 57].

In summary, some of the above label-informed methods [14, 48, 59] perform an indirect representation learning of anomaly score, while other methods [50, 51, 79] mainly rely on predetermined training target(s) to realize the score discrepancy between the normal and abnormal data. Our proposed Overlap loss adaptively achieves the score discrimination from a distribution view, thus alleviating the need to define hyperparameter(s) as anomaly score target(s) in model training.

Distribution Overlap. Our idea is inspired by some recent studies of out-of-distribution (OOD) or multi-classification tasks in the CV field, whereas they usually consider the overlap of class distribution only as a measurement to describe the characteristics of datasets or to evaluate model quality [17, 26, 27, 40]. The overlap of per-dimensional mean and variance [26] is compared among CIFARs [30], LSUN [68], and SVHN [46] datasets, which measures the similarity of the distribution of normalized pixel values. Distribution overlap of PI width is used as a description of the effectiveness of proposed methods [40], where more effective methods should be able to separate the PI width between in-distribution (InD) and out-of-distribution (OOD) samples. IGEOOD [17] score overlap illustrates how each of the presented techniques contributes towards separating InD and OOD samples. RevIN [27] verifies its effectiveness by comparing the distribution discrepancy between training and testing data of a variable on each step of the sequential process in RevIN. Quantitative analysis of distribution overlap is used for describing the shifts of feature statistics between training source domains and unseen testing domains, where the proposed model has less shift for different data distribution.

Considering research more closely related to our work, Magnet loss [56] is proposed to achieve local discrimination by penalizing class distribution overlap, in order to realize explicit modeling of the distributions of different classes in representation space. Motivated by the entropy minimization principle [21], the MA-DNN [12] method minimizes the model entropy (similar to minimizing class distribution overlap) in the feature space and penalizing inconsistent network predictions at the class level.

Nevertheless, these two methods are mainly devised for distance metric learning (DML) that presents optimization in the representation space. We propose to directly optimize distribution overlap in the anomaly scoring space to realize adaptive score distribution discrimination in input instances, which is tailored for the weakly-supervised AD problem where there exist data noises and only a limited number of labeled anomalies.

3 METHODOLOGY

3.1 Problem Statement

Assume the training dataset $\mathcal{D} = \left\{\boldsymbol{x}_1^n, \ldots, \boldsymbol{x}_k^n, \left(\boldsymbol{x}_{k+1}^a, y_{k+1}^a\right), \ldots, \left(\boldsymbol{x}_{k+m}^a, y_{k+m}^a\right)\right\}$ collects both unlabeled instances $\mathcal{D}_n = \left\{\boldsymbol{x}_i^n\right\}_{i=1}^k$ and a handful of labeled anomalies $\mathcal{D}_a = \left\{\left(\boldsymbol{x}_j^a, y_j^a\right)\right\}_{j=1}^m$, where $\boldsymbol{x} \in \mathbb{R}^d$ represents the input feature and y_j^a is the label of identified anomalies. Usually, we have $m \ll k$, since only limited prior knowledge of anomalies is available. This is often considered as the weakly-supervised AD problem [80] and has been studied in several recent works [25, 50, 51, 79]. Given such a dataset, our goal is to train a model, to effectively assign higher anomaly score for the abnormal data.

3.2 Overview of the Proposed Overlap Loss

Overlap loss first employs a *Score Distribution Estimator* for estimating the unknown probability density function (PDF) of the output anomaly scores in neural networks and then conducts *Overlap Area Calculation* between the anomaly score distributions of the unlabeled samples and labeled anomalies. Finally, Overlap loss minimizes the calculated overlap area of score distributions to provide the gradient for backpropagation in neural networks. The proposed Overlap loss fulfills the following properties: (i) the boundness of training loss for better convergence in the model training. (ii) eliminating explicit training target of anomaly score (e.g., constant or margin hyperparameter(s)) to enhance the model adaptability to different datasets. (iii) optimizing the entire anomaly score distribution, instead of pointwise optimization between the estimated anomaly scores and their corresponding targets.

3.3 Overlap Loss for Score Distribution Discrimination

In the following subsections, we illustrate two main parts of proposed Overlap loss: *Score Distribution Estimator* and *Overlap Area Calculation*, along with their corresponding basic ideas and challenges, as complements to our final solutions.

3.3.1 Score Distribution Estimator. Instead of pointwise optimization of the output anomaly scores, here we consider optimizing the anomaly score from a distribution view. Let $Q \in \mathbb{R}^M$ be the hidden representation space, an end-to-end anomaly scoring network $\phi(\cdot;\Theta) : \boldsymbol{x} \mapsto \mathbb{R}$ can be defined as a combination of a feature representation learner $\psi(\cdot;\Theta_t) : \boldsymbol{x} \mapsto Q$ and an anomaly scoring function $\eta(\cdot;\Theta_s) : Q \mapsto \mathbb{R}$, in which $\Theta = \{\Theta_t, \Theta_s\}$. If we denote the anomaly score of normal data as $\phi(\boldsymbol{x}^n;\Theta) = s_n$ and that of abnormal data as $\phi(\boldsymbol{x}^a;\Theta) = s_a$, a density estimator $f(\cdot)$ is then applied to estimate the PDFs of both $\boldsymbol{s}_n$ and $\boldsymbol{s}_a$ in a training batch.

A straightforward idea is to employ a prior distribution, e.g., the Gaussian distribution, as the score distribution estimator. Gaussian distribution inherits several good properties. For instance, the intersection point c used for estimating the score distribution overlap in Figure 2a can be calculated by the following formula[2]:

$$c = \frac{\mu_a\sigma_n^2 - \sigma_a\left(\mu_n\sigma_a + \sigma_n\sqrt{(\mu_n-\mu_a)^2 + 2\left(\sigma_n^2-\sigma_a^2\right)\log\left(\frac{\sigma_n}{\sigma_a}\right)}\right)}{\sigma_n^2-\sigma_a^2} \tag{1}$$

The main challenge of this basic idea is that the number of labeled anomalies is usually too small to satisfy the Gaussian distribution assumption according to the central limit theorem [8], while enforcing the anomaly scores to follow this Gaussian prior would limit the representational ability of neural networks and further distort the anomaly scoring space, resulting in suboptimal performance.

To address the above challenges, we employ a score distribution estimator that is capable of estimating *arbitrary* distribution of output anomaly scores. In this paper, we use the non-parametric Kernel Density Estimation (KDE) method for estimating the arbitrary anomaly score distribution that may be caused by the scarcity of labeled data or the anomaly contamination in the unlabeled data. Actually, other differentiable density estimators can also be applied into our proposed Overlap loss.

If we denote the output anomaly score as $\phi(\boldsymbol{x};\Theta) = s$, the empirical cumulative distribution function (ECDF) can be defined as:

$$\hat{F}_N(s) = \frac{1}{N}\sum_{i=1}^{N}\mathbf{1}_{s_i \le s} \tag{2}$$

where $\mathbf{1}$ is the indicator function and N is the number of partitions. $\hat{F}_N(s)$ is an unbiased estimator [13] of the cumulative distribution function (CDF) $F(s)$, and can be further used for estimating the PDF by the following equation:

$$\begin{aligned}\hat{f}(s) &= \lim_{h\to 0}\frac{\hat{F}_N(s+h) - \hat{F}_N(s-h)}{2h} \\ &\approx \frac{1}{2Nh}\sum_{i=1}^{N}\left(\mathbf{1}_{s_i \le s+h} - \mathbf{1}_{s_i \le s-h}\right) \\ &= \frac{1}{2Nh}\sum_{i=1}^{N}\left(\mathbf{1}_{s-h\le s_i\le s+h}\right) = \frac{1}{Nh}\sum_{i=1}^{N}\frac{1}{2}\mathbf{1}\left(\frac{|s-s_i|}{h} \le 1\right)\end{aligned} \tag{3}$$

where h is the bandwidth. If we denote the kernel function as $K(s) = \frac{1}{2}\mathbf{1}(s \le 1)$, the estimated PDF can be rewritten as:

$$\hat{f}(s) = \frac{1}{Nh}\sum_{i=1}^{N}K\left(\frac{|s-s_i|}{h}\right) = \frac{1}{Nh}\sum_{i=1}^{N}K\left(\frac{s-s_i}{h}\right) \tag{4}$$

where $K(\cdot)$ is symmetric. We use Gaussian kernel in KDE, i.e., $K(s;h) \propto \exp\left(-\frac{s^2}{2h^2}\right)$, for estimating the unknown PDFs of anomaly scores, where the PDFs are further utilized to calculate the overlap area of score distributions, as described in the following subsection.

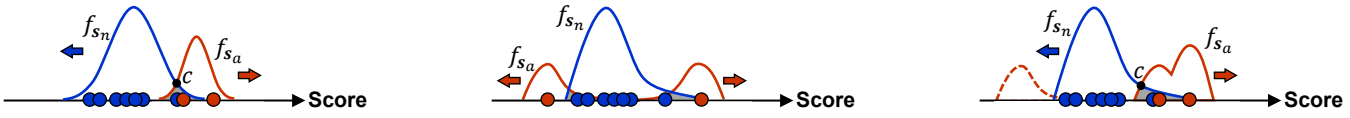
3.3.2 Overlap Area Calculation. Once we obtain the estimated score distributions (i.e., PDFs), the score distribution overlap can be calculated as the overlap area of PDFs between normal samples and the abnormal ones.

An optional method is to directly use the integral to approximate the score distribution overlap, as illustrated in Eq. 5. The overlap area between the PDFs of $\boldsymbol{s}_n$ and $\boldsymbol{s}_a$ is formulated as the integral of the one with the smaller probability density:

$$O(\boldsymbol{s}_n, \boldsymbol{s}_a) = \int_{\min(\boldsymbol{s}_n,\boldsymbol{s}_a)}^{\max(\boldsymbol{s}_n,\boldsymbol{s}_a)} \min\left(\hat{f}_{\boldsymbol{s}_n}(t), \hat{f}_{\boldsymbol{s}_a}(t)\right) dt \tag{5}$$

The main challenge of the above basic idea is that such a method *does not* necessarily guarantee a correct gradient update direction for anomaly scores, as illustrated in Figure 2b. The neural networks could minimize the overlap area of the score distributions, while mistakenly assigning lower anomaly scores for the anomalies (e.g., the left side in the score distribution of the anomalies in Figure 2b) instead of the normal ones. This problem can be remedied through the multi-task learning form by combining Eq.5 with a ranking loss term [65], as shown in Eq.6. However, although it ensures the order in anomaly scores, i.e., the anomaly scores of abnormal data should be further ranked higher than that of normal data, such a method may suffer from the difficult optimization problem in

[2] https://stats.stackexchange.com/questions/103800/calculate-probability-area-under-the-overlapping-area-of-two-normal-distributi



(a) Gaussian score distribution overlap (b) Arbitrary score distribution overlap (c) Proposed score distribution overlap

Figure 2: Anomaly score distribution overlaps. (a) The prior assumption of Gaussian distributions limits the representational ability of neural networks. (b) Overlap of arbitrary score distributions leads to the disorder in anomaly scores. (c) The proposed Overlap loss minimizes the overlap area of arbitrary score distributions while ensuring correct order in anomaly scores.

multi-task learning, sometimes leading to worse performance and data inefficiency compared to learning tasks individually [52, 69].

$$O\left(\boldsymbol{s}_n, \boldsymbol{s}_a\right) = \int_{\min(s_n, s_a)}^{\max(s_n, s_a)} \min\left(\hat{f}_{s_n}(t), \hat{f}_{s_a}(t)\right) dt + \max\left(0, \boldsymbol{s}_n - \boldsymbol{s}_a\right) \tag{6}$$

Our proposed Overlap loss aims to calculate the overlap area of *arbitrary* score distributions while ensuring the correct *order* in anomaly scores. We manage to acquire the intersection point c of these arbitrary score distributions (see Figure 2c), and the score distribution overlap between $\boldsymbol{s}_n$ and $\boldsymbol{s}_a$ in a training batch can be further formulated as Eq.7, where $\hat{F}_{s_n}(\cdot)$ and $\hat{F}_{s_a}(\cdot)$ are the estimated CDF of normal and abnormal data, respectively.

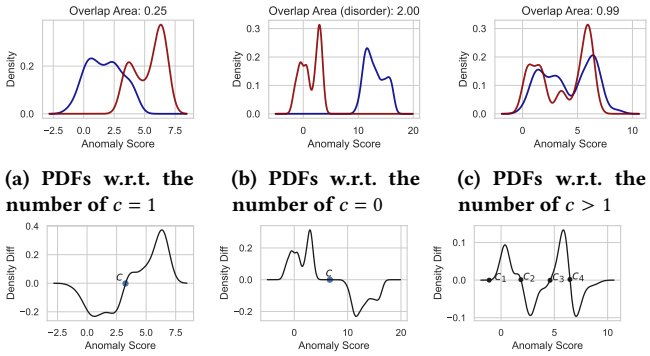
$$O\left(\boldsymbol{s}_n, \boldsymbol{s}_a\right) = P\left(\boldsymbol{s}_n > c\right) + P\left(\boldsymbol{s}_a < c\right) = 1 - \hat{F}_{s_n}(c) + \hat{F}_{s_a}(c) \tag{7}$$

As shown in Figure 2c, the Overlap loss formulated in Eq.7 guarantees the order in output anomaly scores. A small overlap area with correct score order means a close to zero loss of $O\left(\boldsymbol{s}_n, \boldsymbol{s}_a\right)$. If the anomaly scores of abnormal data are smaller than that of normal data, $O\left(\boldsymbol{s}_n, \boldsymbol{s}_a\right)$ would penalize this disorder and be close to 2, since both $P\left(\boldsymbol{s}_n > c\right)$ and $P\left(\boldsymbol{s}_a < c\right)$ are close to 1, respectively. Moreover, $O\left(\boldsymbol{s}_n, \boldsymbol{s}_a\right)$ is naturally bounded to $[0, 2]$ due to the property of PDF.

However, for two arbitrary score distributions, we can not directly calculate the intersection point c by the formula suitable for Gaussian distribution in Eq.1. Instead, we acquire the intersection point c as the corresponding x value of the non-zero element of d_k^s in Eq.8 for the arbitrary score distribution scenario, where s_k is generated by the arithmetic sequence $s_k = \min\left(\boldsymbol{s}_n, \boldsymbol{s}_a\right) + (k-1)\frac{\max(s_n, s_a) - \min(s_n, s_a)}{N}$. In other words, we compare the PDF differences between two adjacent points of the score distributions $\boldsymbol{s}_a$ and $\boldsymbol{s}_n$, as shown in Figure 3.

$$d_k^s = \operatorname{sgn}\left(\hat{f}_{s_a}\left(s_{k+1}\right) - \hat{f}_{s_n}\left(s_{k+1}\right)\right) - \operatorname{sgn}\left(\hat{f}_{s_a}\left(s_k\right) - \hat{f}_{s_n}\left(s_k\right)\right), \quad \textit{where} \quad k = 1, \ldots, N \tag{8}$$

Figure 3 shows toy examples of calculating the intersection point(s) c. For most cases where there is only one intersection point between $\hat{f}_{s_n}(\cdot)$ and $\hat{f}_{s_a}(\cdot)$, c is regarded as the x value of the sign change point of PDF differences, as shown in Figure 3a and 3d. Even if the two score distributions are far apart (see Figure 3b), we could still extend the x range of their PDFs and acquire c, as shown in Figure 3e. It is worth noting that for the case in Figure 3b, Overlap loss would reach its upper bound with an overlap area of 2 for penalizing the disorder in estimated anomaly scores, as illustrated in Eq.7. For the case where there exist multiple intersection points (shown in Figure3c and 3f), we randomly choose one of them as c. We show in the Appendix that the detection performance of this strategy is very close to that of ensembling different intersection points while improving the efficiency of model training.



(a) PDFs w.r.t. the number of $c = 1$ (b) PDFs w.r.t. the number of $c = 0$ (c) PDFs w.r.t. the number of $c > 1$

(d) PDFs diff w.r.t. the number of $c = 1$ (e) PDFs diff w.r.t. the number of $c = 0$ (f) PDFs diff w.r.t. the number of $c > 1$

Figure 3: Calculation of intersection point(s) for arbitrary anomaly score distributions of $f_{s_n}(\cdot)$ (blue) and $f_{s_a}(\cdot)$ (red). (a)~(c) correspond to the situations of one intersection point, no intersection point, and multiple intersection points, respectively. (d)~(f) are their corresponding PDF differences.

After that, the integral of CDF $F_s(c)$ can be approximated via the trapezoidal rule [11] in Eq.9, where Δs_k is adjusted based on the intersection point as $\Delta s_k = [c - \min\left(\boldsymbol{s}_n, \boldsymbol{s}_a\right)]/N$.

$$\hat{F}_s(c) = \int_{-\infty}^{c} \hat{f}_s(t)dt = \int_{\min(s_n, s_a)}^{c} \hat{f}_s(t)dt \approx \sum_{k=1}^{N} \frac{\hat{f}_s\left(s_k\right) + \hat{f}_s\left(s_{k+1}\right)}{2} \Delta s_k \tag{9}$$

Based on the above notations, we define the proposed Overlap loss as follows:

$$\begin{aligned} \mathcal{L}_{\text{Overlap}}\left(\boldsymbol{x} \mid \Theta\right) &= O\left(\boldsymbol{s}_n, \boldsymbol{s}_a\right) + \lambda \|\Theta\|_2^2 \\ &= 1 - \sum_{k=1}^{N} \frac{\hat{f}_{s_n}\left(s_{n,k}\right) + \hat{f}_{s_n}\left(s_{n,k+1}\right)}{2} \Delta s_{n,k} + \\ &\quad \sum_{k=1}^{N} \frac{\hat{f}_{s_a}\left(s_{a,k}\right) + \hat{f}_{s_a}\left(s_{a,k+1}\right)}{2} \Delta s_{a,k} + \lambda \|\Theta\|_2^2 \end{aligned} \tag{10}$$

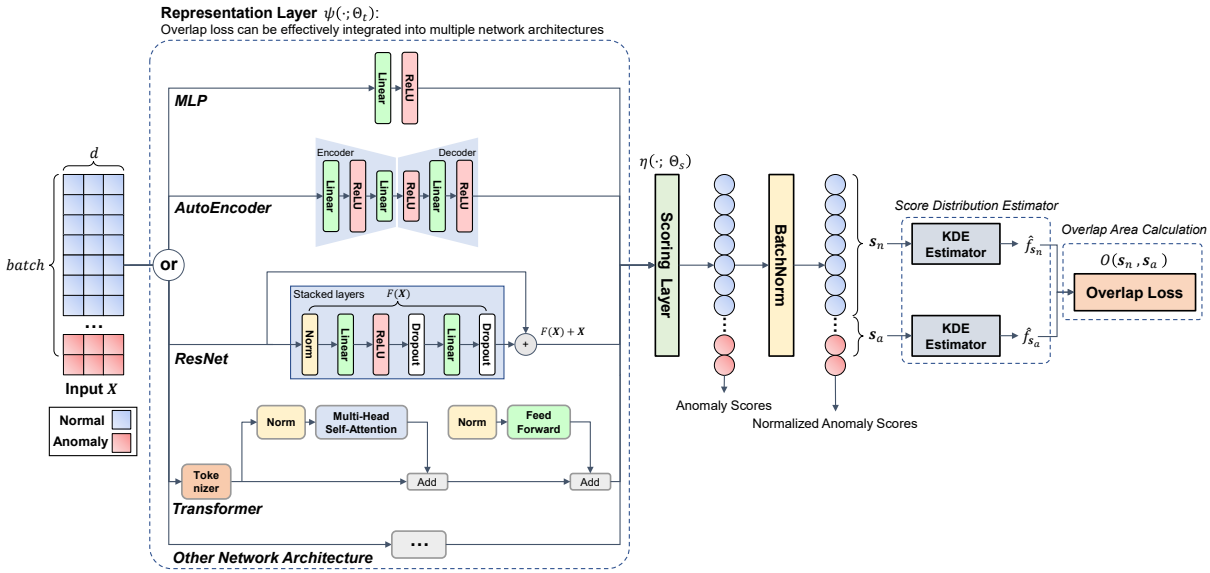


Figure 4: AD model instantiated by the proposed Overlap loss, which consists of a representation layer $\psi(\cdot;\Theta_t)$ and a scoring layer $\eta(\cdot;\Theta_s)$ with batch normalization. The output anomaly scores are used for estimating the score distributions (PDFs) of normal samples $\hat{f}_{s_n}(\cdot)$ and that of anomalies $\hat{f}_{s_a}(\cdot)$ via the KDE estimators. Finally, the calculated overlap area of anomaly score distributions is minimized.

where $\lambda\|\Theta\|_2^2$ is a standard weight decay regularizer controlled by the hyperparameter λ.

3.4 Network Architecture

Overlap loss is instantiated into an end-to-end neural network that consists of a feature representation layer $\psi(\cdot;\Theta_t)$ and a scoring layer $\eta(\cdot;\Theta_s)$. The BatchNorm layer is applied after the scoring layer to normalize the output anomaly scores. After that, score distributions of both normal data and anomalies are estimated by the KDE estimators, where their score distribution overlap is further calculated via the proposed Overlap loss, as shown in Figure 4.

We point out that the proposed Overlap loss can be effectively integrated into multiple popular network architectures, including the widely-used MLP and AutoEncoder in AD tasks, and some cutting-edge architectures like ResNet and Transformer in the classification tasks. Algorithm 1 provides detailed steps of instantiated models based on our proposed Overlap loss.

Algorithm 1: AD model instantiated by the Overlap loss

Input: Unlabeled instances $\mathcal{D}_n = \{\boldsymbol{x}_i^n\}_{i=1}^k$, a limited number of identified anomalies $\mathcal{D}_a = \{(\boldsymbol{x}_j^a, y_j^a)\}_{j=1}^m$
Output: Anomaly scores $\boldsymbol{s}$
Initialize network parameters of both feature representation layer Θ_t and scoring layer Θ_s
for *epoch=1:*n_{epoch} **do**
  for *batch=1:*n_{batch} **do**
    Randomly sample unlabeled instances $\boldsymbol{x}_{batch}^n$ from $\mathcal{D}_n$ and labeled anomalies $\boldsymbol{x}_{batch}^a$ from $\mathcal{D}_a$
    (1) Acquire the anomaly scores with $\phi(\boldsymbol{x}_{batch}^n;\Theta) = \boldsymbol{s}_{batch}^n$ and $\phi(\boldsymbol{x}_{batch}^a;\Theta) = \boldsymbol{s}_{batch}^a$
    (2) Use the KDE method to estimate the PDF $\hat{f}_{\boldsymbol{s}_{batch}^n}(\cdot)$ and $\hat{f}_{\boldsymbol{s}_{batch}^a}(\cdot)$ of the output anomaly scores
    (3) Calculate the intersection point c by Eq. 8
    (4) Approximate the CDFs by the trapezoidal rule in Eq. 9
    (5) Calculate and minimize the score distribution overlap $O\left(\boldsymbol{s}_{batch}^n, \boldsymbol{s}_{batch}^a\right)$ via Eq. 10
    (6) Perform backpropagation and update network parameters $\Theta = \{\Theta_t, \Theta_s\}$
  end
end
Output anomaly scores via learned scoring function $\phi(\boldsymbol{x};\Theta) = \boldsymbol{s}$

4 EXPERIMENTS

4.1 Experiment Setting

Datasets. We apply 25 publicly available real-world datasets for model evaluation. These datasets include several domains such as disease diagnosis, speech recognition, and image identification. For each dataset, 70% data is split as the training set and the remaining 30% as the testing set, where the same proportion of anomalies is kept by the stratified sampling. We discuss the model performance in Section 4.3.1 w.r.t. different ratios of labeled anomalies to all true anomalies $\gamma_l = m/(k+m)$ in the training set, where m labeled anomalies are sampled from the entire anomaly data and the rest of k instances remain unlabeled.

Baselines. We compare the proposed Overlap loss based AD models with the following baselines. The Unlearning method [14] is not included here since it is originally proposed for the time-series task, while we explore the Minus loss of the Unlearning method

Table 1: Dataset description.

Dataset	N	D	#anomalies	#anomaly ratio (%)
ALOI	49534	27	1508	3.04
annthyroid	7200	6	534	7.42
Cardiotocography	2114	21	466	22.04
fault	1941	27	673	34.67
http	567498	3	2211	0.39
landsat	6435	36	1333	20.71
letter	1600	32	100	6.25
magic.gamma	19020	10	6688	35.16
mammography	11183	6	260	2.32
mnist	7603	100	700	9.21
musk	3062	166	97	3.17
optdigits	5216	64	150	2.88
PageBlocks	5393	10	510	9.46
pendigits	6870	16	156	2.27
satellite	6435	36	2036	31.64
satimage-2	5803	36	71	1.22
shuttle	49097	9	3511	7.15
skin	245057	3	50859	20.75
SpamBase	4207	57	1679	39.91
speech	3686	400	61	1.65
thyroid	3772	6	93	2.47
vowels	1456	12	50	3.43
Waveform	3443	21	100	2.90
Wilt	4819	5	257	5.33
yeast	1484	8	507	34.16

in Section 4.4. We do not include the results of DAGMM [81] for comparison as it may not converge on some datasets. We exclude the semi-supervised Dual-MGAN [35] method for comparison since it is too computationally expensive.

(i) **Typical AD algorithms**, including unsupervised shallow methods like Iforest [39] and ECOD [36], and DL-based representation learning methods like unsupervised DeepSVDD [58] and semi-supervised DeepSAD [59].

- **Iforest** [39]. An ensemble of binary trees defines the anomaly score as the closeness of an individual instance to the root.
- **ECOD** [36]. A parameter-free method that estimates the empirical cumulative distribution of input features and regards tail probabilities as the anomaly score.
- **DeepSVDD** [58]. A neural network based model that describes the anomaly score as the distance of transformed embedding to the center of the hypersphere.
- **GANomaly** [3]. A GAN-based method that defines the reconstruction error of the input instance as the anomaly score.
- **DeepSAD** [59]. A deep semi-supervised one-class method that improves the unsupervised DeepSVDD.
- **REPEN** [48]. A neural network based model that leverages transformed low-dimensional representation for random distance-based detectors.

(ii) **MLP based AD algorithms**, including weakly-supervised DevNet [51] and PReNet [50] models that additionally leverage a handful of labeled anomalies to improve detection performance.

- **DevNet** [51]. A neural network based model that uses a prior probability to enforce the statistical deviation score of input instances.
- **PReNet** [50]. A neural network based model that defines a two-stream ordinal regression to learn the relation of instance pairs.

(iii) **AutoEncoder based AD algorithms**, including weakly-supervised method FEAWAD [79] and its supervised version.

- **FEAWAD (weak)** [79]. A neural network based model that incorporates the network architecture of DAGMM [81] with the deviation loss of DevNet.
- **FEAWAD (sup)**. A baseline that applies the same network architecture as FEAWAD, but uses the binary cross entropy loss to train the model.

(iv) **Supervised ResNet classifier** tailored for tabular data.

- **ResNet** [19]. ResNet-like architecture turns out to be a strong baseline that is often missing in prior tabular AD tasks.

(v) **Supervised Transformer classifier** tailored for tabular data.

- **FTTransformer** [19]. A Transformer architecture implements with Feature Tokenizer. FTTransformer has been proven to be better than other DL solutions on tabular tasks.

4.2 Training Details

For unsupervised baselines, Iforest, ECOD, and DeepSVDD are built using the PyOD [77] library[3]. Labeled anomalies are combined with unlabeled data for constructing the validation set, in order to tune the hyperparameters of these unsupervised methods via the grid search method, since tuning their hyperparameters on a small validation set often yields better performance than using the default settings [61]. Table 2 shows the hyperparameter grids, where ECOD is not considered since it is a parameter-free method.

Table 2: Hyperparameter grid of the unsupervised models.

Model	Hyperparameter
Iforest	n_estimators: [10, 50, 100, 500]
DeepSVDD	epochs: [20, 50, 100, 200]

We replace the convolutional layer in the original GANomaly with the dense layer for evaluating it on the tabular data, where the hidden size of the encoder-decoder-encoder structure of GANomaly is set to half of the input dimension. We realize the PReNet in PyTorch[4] as we do not find the open-source codes, and set the hyperparameters in PReNet according to its original paper. Other models are built based on their corresponding source codes.

For the proposed Overlap loss based AD models, we use the SGD optimizer with 0.001 learning rate and 0.7 momentum. The weight decay λ is set to 0.01. The bandwidth h in the KDE method is set to 1. The N in Eq.8~10 is set to 1000 by default. We train the Overlap loss based MLP and AutoEncoder models (namely MLP-Overlap and AE-Overlap) for 20 epochs, where batch size of 256 is used. For ResNet and FTTransformer architectures, we train the Overlap loss based models (namely ResNet-Overlap and FTTransformer-Overlap) 100 training epochs just as their original paper. All the experiments are run on a Tesla V100 GPU accelerator.

Metrics. We evaluate the above models by two metrics: the AUC-ROC (Area Under Receiver Operating Characteristic Curve) and the AUC-PR (Area Under Precision-Recall Curve) values. Furthermore,

[3]https://pyod.readthedocs.io/en/latest/
[4]https://pytorch.org/

Table 3: Average performance over 25 real-world datasets. Each experiment is repeated 5 times. Architecutre column specifies which network architecture overlap loss is based on. Category column refers to the algorithm categories in their original papers, including unsupervised (Unsup), weakly-supervised (Weak), semi-supervised (Semi) and fully-supervised (Sup) categories. γ_l stands for the ratio of labeled anomalies to all true anomalies in the training set. Δ Perf. shows the relative improvement of overlap loss based models over their corresponding counterparts. ***, ** and * denote statistical significance at 1%, 5% and 10% of Wilcoxon signed rank test, respectively. The best results are in bold. Please see the Appendix for complete experimental results.

(a) AUC-ROC results of model comparison.

Architecture	Model	Category	γ_l = 5%		γ_l = 10%		γ_l = 20%	
			AUC-ROC	Δ Perf.	AUC-ROC	Δ Perf.	AUC-ROC	Δ Perf.
Typical	Iforest	Unsup	0.737±0.187	/	0.737±0.187	/	0.737±0.187	/
	ECOD	Unsup	0.701±0.208	/	0.701±0.208	/	0.701±0.208	/
	DeepSVDD	Unsup	0.504±0.028	/	0.504±0.028	/	0.504±0.028	/
	GANomaly	Semi	0.655±0.162	/	0.648±0.153	/	0.665±0.152	/
	DeepSAD	Semi	0.823±0.142	/	0.859±0.136	/	0.888±0.129	/
	REPEN	Weak	0.810±0.166	/	0.832±0.165	/	0.848±0.163	/
MLP	DevNet	Weak	0.842±0.148	+0.57%	0.861±0.135	+2.16%	0.873±0.129	+2.40%
	PReNet	Weak	0.846±0.146	+0.18%	0.866±0.132	+1.61%	0.876±0.127	+1.97%
	MLP-Overlap (ours)	Weak	**0.847±0.145**	/	**0.880±0.132**	/	**0.893±0.133**	/
AutoEncoder	FEAWAD	Sup	0.771±0.211	+11.72%***	0.849±0.133	+4.34%***	0.876±0.133	+2.42%***
	FEAWAD	Weak	0.808±0.154	+6.60%***	0.848±0.145	+4.51%***	0.876±0.129	+2.43%***
	AE-Overlap (ours)	Weak	**0.862±0.144**	/	**0.886±0.137**	/	**0.897±0.132**	/
ResNet	ResNet	Sup	0.651±0.158	+28.48%***	0.736±0.124	+19.74%***	0.816±0.127	+11.03%***
	ResNet-Overlap (ours)	Weak	**0.836±0.146**	/	**0.882±0.134**	/	**0.906±0.122**	/
Transformer	FTTransformer	Sup	0.827±0.159	+3.00%**	0.859±0.146	+1.80%	0.889±0.129	+1.23%
	FTTransformer-Overlap (ours)	Weak	**0.851±0.138**	/	**0.874±0.130**	/	**0.900±0.127**	/

(b) AUC-PR results of model comparison.

Architecture	Model	Category	γ_l = 5%		γ_l = 10%		γ_l = 20%	
			AUC-PR	Δ Perf.	AUC-PR	Δ Perf.	AUC-PR	Δ Perf.
Typical	Iforest	Unsup	0.389±0.295	/	0.389±0.295	/	0.389±0.295	/
	ECOD	Unsup	0.315±0.239	/	0.315±0.239	/	0.315±0.239	/
	DeepSVDD	Unsup	0.147±0.120	/	0.147±0.120	/	0.147±0.120	/
	GANomaly	Semi	0.297±0.191	/	0.296±0.195	/	0.306±0.201	/
	DeepSAD	Semi	0.506±0.253	/	0.601±0.275	/	0.675±0.284	/
	REPEN	Weak	0.560±0.300	/	0.603±0.308	/	0.639±0.306	/
MLP	DevNet	Weak	0.606±0.311	+2.89%	0.626±0.307	+7.75%**	0.652±0.305	+6.74%*
	PReNet	Weak	0.612±0.305	+1.82%	0.638±0.307	+5.67%*	0.660±0.303	+5.49%
	MLP-Overlap (ours)	Weak	**0.623±0.291**	/	**0.674±0.286**	/	**0.696±0.288**	/
AutoEncoder	FEAWAD	Sup	0.509±0.269	+28.04%***	0.620±0.270	+12.05%***	0.678±0.270	+5.17%**
	FEAWAD	Weak	0.596±0.286	+9.29%***	0.645±0.293	+7.71%***	0.682±0.283	+4.56%**
	AE-Overlap (ours)	Weak	**0.652±0.290**	/	**0.695±0.294**	/	**0.713±0.296**	/
ResNet	ResNet	Sup	0.401±0.241	+56.30%***	0.483±0.224	+44.81%***	0.598±0.235	+23.92%***
	ResNet-Overlap (ours)	Weak	**0.627±0.297**	/	**0.699±0.289**	/	**0.742±0.283**	/
Transformer	FTTransformer	Sup	0.594±0.299	+5.50%*	0.644±0.308	+6.61%*	0.691±0.305	+5.65%*
	FTTransformer-Overlap (ours)	Weak	**0.627±0.277**	/	**0.686±0.282**	/	**0.730±0.285**	/

we apply the pairwise Wilcoxon signed rank test [66] to examine the significance of proposed methods against its competitors.

4.3 Experimental Results

4.3.1 Model Performance. Table 3 shows the average model performance over 25 real-world datasets, and we report the full results in the Appendix of supplementary materials. Above all, we verify the effectiveness of the proposed Overlap loss on various network architectures, including MLP, AutoEncoder, ResNet, and Transformer. The Overlap loss based AD models generally outperform corresponding baselines in terms of AUC-ROC and AUC-PR w.r.t. the ratios of labeled anomalies γ_l = 5%, γ_l = 10% and γ_l = 20%.

Specifically, experimental results show that the MLP-Overlap achieves a relative improvement Δ Perf. of AUC-ROC over its counterpart DevNet 0.57% and PReNet 0.18%, and Δ Perf. of AUC-PR over DevNet 2.89% and PReNet 1.82% w.r.t. γ_l = 5%. These results indicate that compared to the current state-of-the-art weakly-supervised AD methods, Overlap loss is still more effective when only a handful of labeled anomalies (say 5% labeled anomalies) are available in the training process, considering that such limited label information would bring challenges for estimating anomaly score distribution. Besides, we show that end-to-end weakly-supervised AD methods, including DevNet, PReNet, and our MLP-Overlap,

Table 4: Experimental results of ablation studies. Overlap-Gaussian refers to the basic method mentioned in Section 3.3.1. Overlap-Arbitrary refers to the basic method of Eq.5. Overlap-Ranking isolates the ranking loss in Eq.6. Overlap-Combined corresponds to the combined loss form of both Overlap-Arbitrary and Overlap-Ranking as illustrated in Eq.6. Overlap-Proposed refers to the final solution in this paper.

(a) AUC-ROC results of ablation studies.

Method	γ_l = 5%					γ_l = 10%					γ_l = 20%				
	VAE	**MLP**	**AE**	**ResNet**	**FTT**	**VAE**	**MLP**	**AE**	**ResNet**	**FTT**	**VAE**	**MLP**	**AE**	**ResNet**	**FTT**
Overlap-Gaussian	0.539	/	/	/	/	0.540	/	/	/	/	0.541	/	/	/	/
Overlap-Arbitrary	/	0.496	0.531	0.493	0.521	/	0.498	0.543	0.534	0.482	/	0.502	0.516	0.541	0.483
Overlap-Ranking	/	0.810	0.822	0.807	**0.862**	/	0.845	0.857	0.855	**0.888**	/	0.873	0.874	0.890	**0.906**
Overlap-Combined	/	0.843	0.854	0.820	0.610	/	**0.881**	0.885	0.874	0.602	/	**0.898**	**0.901**	**0.908**	0.589
Overlap-Proposed	/	**0.847**	**0.862**	**0.836**	0.851	/	0.880	**0.886**	**0.882**	0.874	/	0.893	0.897	0.906	0.900

(b) AUC-PR results of ablation studies.

Method	γ_l = 5%					γ_l = 10%					γ_l = 20%				
	VAE	**MLP**	**AE**	**ResNet**	**FTT**	**VAE**	**MLP**	**AE**	**ResNet**	**FTT**	**VAE**	**MLP**	**AE**	**ResNet**	**FTT**
Overlap-Gaussian	0.159	/	/	/	/	0.158	/	/	/	/	0.158	/	/	/	/
Overlap-Arbitrary	/	0.349	0.376	0.341	0.356	/	0.376	0.416	0.407	0.339	/	0.397	0.410	0.426	0.352
Overlap-Ranking	/	0.535	0.558	0.539	0.623	/	0.605	0.620	0.625	**0.687**	/	0.657	0.657	0.691	**0.731**
Overlap-Combined	/	**0.624**	0.642	0.604	0.418	/	**0.682**	**0.695**	0.686	0.439	/	**0.708**	**0.719**	0.741	0.459
Overlap-Proposed	/	0.623	**0.652**	**0.627**	**0.627**	/	0.674	**0.695**	**0.699**	0.686	/	0.696	0.713	**0.742**	0.730

statistically outperform those unsupervised (e.g. Iforest) or semi-supervised representational learning (e.g., DeepSAD) AD methods, since end-to-end anomaly score learning can leverage the data much more efficiently than the two-step AD approaches [49, 51].

For γ_l = 10%, the relative improvement of Overlap loss based model is more significant, as more labeled anomalies are beneficial for the Overlap loss to estimate more accurate anomaly score distributions, and thus to better measure the score distribution overlap. The MLP-Overlap Δ Perf. of AUC-ROC over DevNet is 2.16% and 1.61% over PReNet, and Δ Perf. of AUC-PR over DevNet is 7.75% and 5.67% over PReNet w.r.t. γ_l = 10%.

Furthermore, we prove the superiority of the proposed Overlap loss on other network architectures such as AutoEncoder and ResNet. For AUC-ROC, the Δ Perf. of AE-Overlap over fully- and weakly-supervised FEAWAD is 11.72% and 6.60%, respectively, and Δ Perf. of ResNet-Overlap over ResNet is 28.48%. In terms of AUC-PR, the Δ Perf. of AE-Overlap over fully- and weakly-supervised FEAWAD is 28.04% and 9.29%, respectively, and Δ Perf. of ResNet-Overlap over ResNet is 56.30%, where all the relative improvements are significant at 1% significance level. Although FTTransformer has been proven to be a strong solution for tabular-based tasks [19], we still observe Δ Perf. of FTTransformer-Overlap over FTTransformer 1.23%~3.0% on AUC-ROC and 5.50%~6.61% on AUC-PR.

4.3.2 Runtime Analysis. We show the model training time in Figure 5. This result shows that ECOD is the fastest algorithm as it treats each feature independently. Both MLP-Overlap and AE-Overlap are faster than their counterparts, since our methods need fewer training epochs while achieving better detection performance. For ResNet and FTTransformer (FTT) architectures, our methods are comparable to or relatively slower than the counterparts. This is mainly due to the fact that Overlap loss requires more training epochs for more complex network architectures (especially for FT-Transformer) than those simple architectures like MLP. Therefore we apply the same training strategy (100 epochs with early stopping) for ResNet and FTTransformer, as well as our Overlap loss based versions ResNet-Overlap and FTTransformer-Overlap. The extra training time is mainly caused by the calculation of Overlap loss, compared to the supervised binary cross entropy loss.

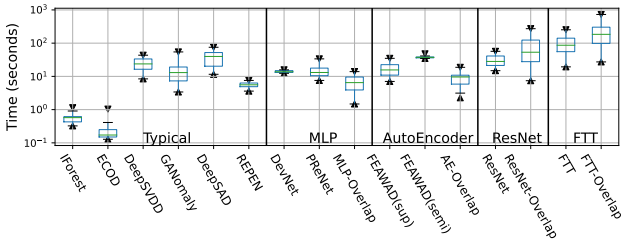


Figure 5: Boxplot of model training time. Different network architectures are separated by vertical lines.

4.3.3 Ablation Study. In Table 4, we report the AUC-ROC and AUC-PR results of several basic methods mentioned in Section 3. We instantiate the basic method of Overlap-Gaussian by replacing the scoring layer $\eta(\cdot;\Theta_s)$ with the VAE [28] structure, where the anomaly scores of normal and abnormal data are sampled from their corresponding Gaussian distribution via the reparameterization trick [28]. The calculated intersection point c of Eq.1 can be used for estimating score distribution overlap via Eq.7.

First, we observe that Overlap-Gaussian has the worst performance. This is because the scarceness of labeled anomalies makes their score distribution often present a certain arbitrariness, whereas the Gaussian assumption is detrimental to the representation of the scoring function. *Second*, the disorder in anomaly scores leads to performance degradation in the Overlap-Arbitrary method. Ranking loss term can be served as an effective way to guarantee the order in

anomaly scores, as the Overlap-Combined method significantly improves the detection performance. *Third*, the proposed Overlap loss (namely Overlap-Proposed) outperforms all basic methods in most cases, since it can effectively estimate arbitrary score distributions of output anomaly scores while avoiding the score disorder problem that occurs in the Overlap-Arbitrary method. Compared to the Overlap-Combined method, Overlap-Proposed achieves better performance, probably because it realizes a unified loss function form, rather than a combination of two different loss parts. Besides, we also observe that the multi-task loss form in the Overlap-Combined method may fail in more complex network architectures like FT-Transformer.

4.4 Further Exploration into AD Loss Functions

While most of the existing research focuses on proposing and evaluating specific models or architectural designs of AD methods [67], we manage to go a step further and directly compare different loss functions in the same network architecture. We introduce the decoupling methods in the Neural Architecture Search (NAS) problem [15, 38], where we mainly concern the design space of loss functions instead of other perspectives like architecture settings [34]. Such an analytical method could eliminate the effects of model configurations such as dropout and activation layers, while fully focusing on the role of loss functions (i.e., training objectives) in the anomaly detection tasks.

Table 5: Summary of loss functions. The No Prior column indicates whether the prior anomaly score target is needed in the corresponding loss functions, e.g., the margin hyperparameter M in the Hinge loss.

Loss	Formula	No Prior
Minus	$\mathcal{L} = \lvert s_n \rvert + \max(0, BND - \lvert s_a \rvert)$	✗
Inverse	$\mathcal{L} = \lvert s_n \rvert + 1/\lvert s_a \rvert$	✓
Hinge	$\mathcal{L} = \max(0, M + s_n - s_a)$	✗
Deviation	$\mathcal{L} = \lvert s_n \rvert + \max(0, M - s_a)$	✗
Ordinal	$\mathcal{L} = \lvert s_{n,n} - \mu_{n,n} \rvert + \lvert s_{a,n} - \mu_{a,n} \rvert + \lvert s_{a,a} - \mu_{a,a} \rvert$	✗
Overlap	Eq.10	✓

We decouple the loss functions in several popular AD models mentioned in Figure 1, including the Minus loss in Unlearning [14], Inverse loss in DeepSAD [59], Hinge loss in REPEN [48], Deviation loss in DevNet [51] and FEAWAD [79], Ordinal loss in PReNet [50], and our proposed Overlap loss, as shown in Table 5. For the consistency of comparison, we replace the original reconstruction error in Minus loss and the Euclidean distance of embedding in Inverse loss with the absolute anomaly score. A network with one-hidden layer of 20 neurons is applied to ensure the comparability of different loss functions. The ReLU activation layer is employed in this network. We train the network for 200 epochs of 256 batch size and use the SGD optimizer with 0.01 learning rate and 0.7 momentum. The weight decay is set to 0.01. The hyperparameter BND in the Minus loss is set to 5, and the hyperparameter M in the Hinge and Deviation loss is set to 5. The anomaly scores in Deviation loss are normalized as Z-Score. Consistent with the original paper, we set $\mu_{n,n}$, $\mu_{n,n}$ and $\mu_{n,n}$ in the Ordinal loss to 0, 4, and 8, respectively.

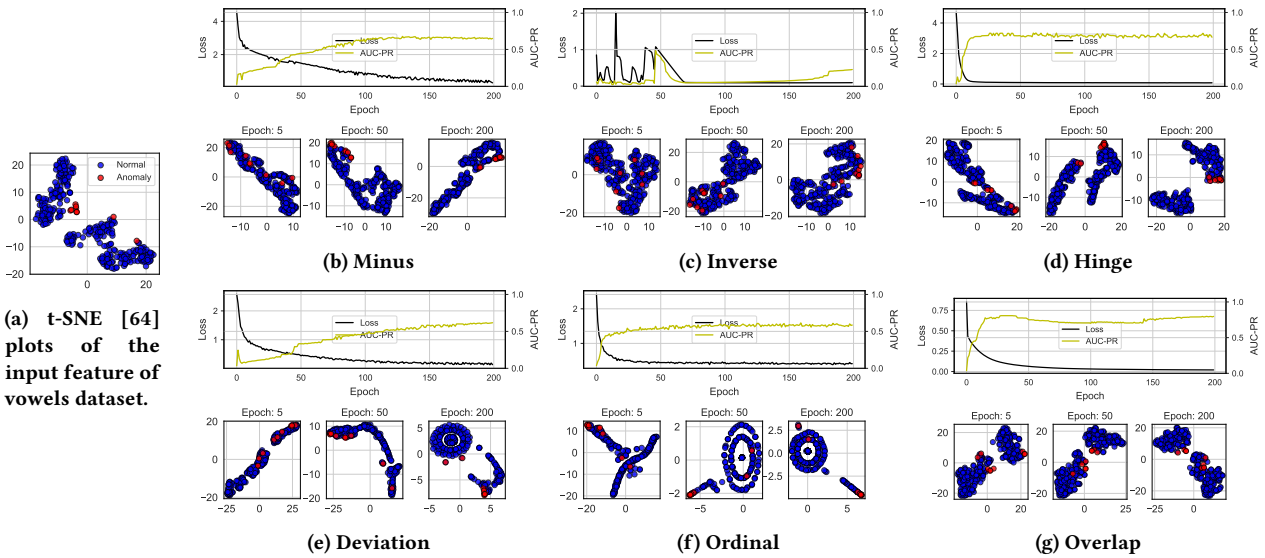
We first investigate different loss functions on 25 real-world datasets and then report their performances in detecting various types of anomalies. Finally, we provide two case studies as a straightforward way to better understand these loss functions in the weakly-supervised scenario.

4.4.1 Exploration of AD Loss Functions on Real-world Datasets. In this subsection, we analyze different loss functions on real-world datasets with respect to the following two perspectives: (i) Embedding transformation. The transformed embedding of input features [3, 4] can be seen as a visualization of representation layer variation for realizing the training objective. (ii) Network Parameter Changes. This is often discussed in the continual learning problem, where drastic changes in network parameters may suffer from the problem of catastrophic forgetting [5, 14, 29, 73]. Similarly, we investigate the network parameter changes of different loss function based AD models when achieving their corresponding training objectives.

Embedding Transformation during Model Training. We take the vowels dataset as an example to demonstrate the embedding transformation in the feature representation layer during the training process, as shown in Figure 6. Similar experimental results can be observed in other real-world datasets, and we provide these results in the Appendix of supplementary materials. Figure 6 indicates that the Deviation loss and Ordinal loss tend to seriously distort the representation of original input data after a few training epochs. This is due to the fact that these two loss functions explicitly guide networks to map the anomaly score of each instance or pair to one or more fixed score constants or a score margin, thus may hinder the diversity of learned representation. In the weakly-supervised scenario, the unlabeled normal samples are contaminated by the unlabeled anomalies, and defining an identical training target for these two types of data would limit the representational ability of the learned models.

Compared to the other loss functions, our proposed Overlap loss generates a relatively mild transformation of the input features. As mentioned before, Overlap loss based AD models can achieve superior detection performance, therefore we speculate that good detection performance does not always require an excessive transformation in the representation space, where the model could only transform the embeddings that have the most impact on the score distribution discrimination and remain more fine-grained information of input data. This conclusion is also consistent with the principle of Occam's razor [9].

Network Parameter Changes. We show the results of network parameter changes on the 25 real-world datasets in Figure 7, where the sum of norms of parameter differences in each layer are calculated between the initialized model and its updated version as $\sum_l \|M - M_0\|_2^2$. The result indicates that compared to the other loss functions, the AD model based on our proposed Overlap loss inherits smaller network parameter changes. This result corresponds to the good properties of Overlap loss, where (i) Overlap loss is naturally bounded, avoiding drastic updating of anomaly scores (and network parameters). (ii) Overlap loss does not require the prior target of anomaly score, therefore reducing unnecessary scoring function updates in the training stage and being capable of adapting to different datasets with minimum adjustment of the score distribution.



(a) t-SNE [64] plots of the input feature of vowels dataset.

(b) Minus (c) Inverse (d) Hinge

(e) Deviation (f) Ordinal (g) Overlap

Figure 6: Training loss along with the AUC-PR performance on testing set of different loss function based AD models, where the vowels dataset is specified for comparison. The transformed embeddings of the input feature are demonstrated, which corresponds to 5, 50, and 200 training epochs, respectively. See the additional results in Appendix.

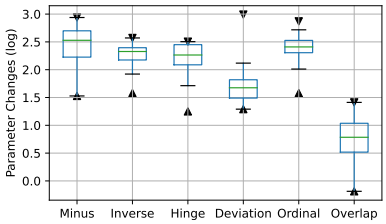


Figure 7: Network parameter changes in the training stage.

4.4.2 Exploration of AD Loss Functions on Different Types of Anomalies. While extensive AD methods have been proven to be effective on real-world datasets, previous studies often neglect to discuss the pros and cons of AD methods regarding specific types of anomalies [18, 62]. In fact, public datasets often consist of a mixture of different types of anomalies. We follow [22, 62] to create realistic synthetic datasets based on the above 25 datasets by injecting the following four types of anomalies to evaluate different loss functions.

- **Local anomalies** refer to the anomalies deviant from their local neighborhoods [10]. GMM procedure [45, 62] is used to generate synthetic normal samples, and then scale the covariance matrix $\hat{\Sigma} = \alpha\hat{\Sigma}$ by a scaling parameter $\alpha = 5$ to generate local anomalies.
- **Global anomalies** are generated from a uniform distribution $\text{Unif}\left(\alpha \cdot \min\left(\boldsymbol{x^k}\right), \alpha \cdot \max\left(\boldsymbol{x^k}\right)\right)$, where the boundaries are defined as the *min* and *max* of an input feature, e.g., k-th feature $\boldsymbol{x^k}$, and $\alpha = 1.1$ controls the outlyingness of anomalies.
- **Dependency anomalies** refer to the samples that do not follow the dependency structure that normal data follows [44], i.e., the input features of dependency anomalies are assumed to be independent of each other. Vine Copula [1] method is applied to model the dependency structure of original data, where the probability density function of generated anomalies is set to complete independence by removing the modeled dependency (see [44]). KDE method estimates the probability density function of features and generates normal samples.
- **Clustered anomalies**, also known as group anomalies [33], exhibit similar characteristics [16, 37]. We scale the mean feature vector of normal samples by $\alpha = 5$, i.e., $\hat{\boldsymbol{\mu}} = \alpha\hat{\boldsymbol{\mu}}$, where α controls the distance between anomaly clusters and the normals, and use the scaled GMM to generate anomalies.

Table 6: Loss function comparison on different types of anomalies generated based on the 25 real-world datasets.

(a) AUC-ROC results of loss comparison.

Loss	Local	Global	Clustered	Dependency
Minus	0.629	0.936	0.996	0.738
Inverse	0.547	0.823	0.937	0.570
Hinge	0.607	0.938	0.997	0.761
Deviation	0.588	0.959	0.990	0.652
Ordinal	0.604	0.954	0.994	0.687
Overlap	0.**742**	**0.981**	**0.998**	**0.847**

(b) AUC-PR results of loss comparison.

Loss	Local	Global	Clustered	Dependency
Minus	0.255	0.822	0.992	0.369
Inverse	0.235	0.647	0.900	0.198
Hinge	0.271	0.853	0.996	0.413
Deviation	0.246	0.851	0.987	0.303
Ordinal	0.247	0.849	0.991	0.327
Overlap	**0.439**	**0.929**	**0.998**	**0.571**

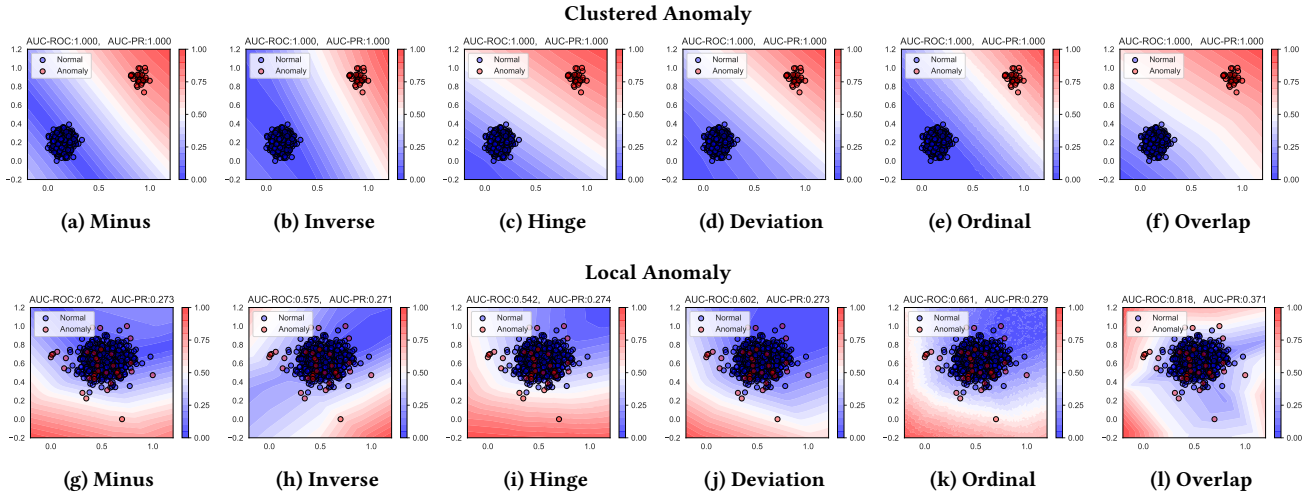


Figure 8: Decision boundaries of different loss functions on the local anomalies. The output anomaly scores are normalized to $[0, 1]$ for comparison. Both AUC-ROC and AUC-PR performances are displayed in the title above each subfigure.

Table 6 shows the results of loss function comparison on different types of anomalies. These results are consistent with the findings in [22], where the other loss functions devised for semi- or weakly-supervised AD algorithms perform relatively poorly on the local and dependency anomalies. Unlike clustered anomalies, the partially labeled anomalies of local and dependency anomalies can not well capture all characteristics of specific types of anomalies, and learning such decision boundaries for seperating normal and abnormal data is often challenging (see Figure 8g~8k). Therefore, the incomplete label information may bias the learning process of these loss functions, which explains their relatively inferior performances compared to the proposed Overlap loss.

In contrast, Overlap loss performs significantly better on local, global, and dependency anomalies, and achieves satisfactory results on clustered anomalies. For example, the average AUC-ROC and AUC-PR performance of Overlap loss on the local anomalies is 0.742 and 0.439, compared to the second-best Minus loss of 0.629 for AUC-ROC and Hinge loss of 0.271 for AUC-PR, respectively. The average AUC-ROC and AUC-PR performance of Overlap loss on the dependency anomalies is 0.847 and 0.571, compared to the second-best Hinge loss of 0.761 and 0.413, respectively.

Such results verify that Overlap loss can effectively leverage the prior knowledge of both partial labels and anomaly types. That is to say, Overlap loss based AD models (e.g., ResNet-Overlap) can achieve superior performance when only a limited number of labeled anomalies are available during the training stage. Furthermore, if one could get access to the valuable prior knowledge of anomaly types [22], Overlap loss can be served as an effective solution to learn the pattern of this specific type (e.g., dependency) of anomalies.

We further investigate two case studies by generating visualized two-dimensional synthetic samples of the above local and clustered anomalies, as shown in Figure 8. The anomaly ratios of these two datasets are set to 5%. The results indicate that all the compared loss functions can correctly detect anomalies for the two-dimensional clustered anomalies (with 1.000 AUC-ROC and AUC-PR). This result can be expected since few labeled clustered anomalies can already represent similar behaviors of the entire clustered anomalies. For the local anomalies, however, we observe most of the compared loss functions perform poorly. In contrast, Overlap loss achieves better detection performance, and successfully learns a suitable decision boundary (see Figure 8l), where the learned decision boundary fits well with the local anomalies that are often overlapped or surrounded by the normal samples.

5 CONCLUSION

In this paper, we propose a novel loss function called Overlap loss for the weakly-supervised AD task. Overlap loss liberates the AD models from the predefined anomaly score targets, e.g., predefined constant or margin hyperparameter(s), thus adapting well to various datasets. By directly optimizing distribution overlap to realize score distribution discrimination, Overlap loss can retain more fine-grained information of input data, and also avoids dramatic changes in network parameters which may lead to overfitting or catastrophic forgetting problem. Extensive experimental results verify that the proposed Overlap loss can be effectively instantiated to different network architectures, including MLP, AutoEncoder, ResNet, and Transformer. Moreover, Overlap loss significantly outperforms other popular AD loss functions on various types of anomalies.

For the future, we plan to improve the optimization process of Overlap loss by leveraging more complex score distribution estimators, such as the Gaussian Mixture Model (GMM) [55]. Besides, we will extend our work to more general scenarios in weakly-supervised AD tasks [80], such as the inaccurate supervision [78] and inexact supervision [32, 63] problems.

REFERENCES

[1] Kjersti Aas, Claudia Czado, Arnoldo Frigessi, and Henrik Bakken. 2009. Pair-copula constructions of multiple dependence. *Insurance: Mathematics and economics* 44, 2 (2009), 182–198.

[2] Charu C Aggarwal. 2017. An introduction to outlier analysis. In *Outlier analysis*. Springer, 1–34.

[3] Samet Akcay, Amir Atapour-Abarghouei, and Toby P Breckon. 2018. Ganomaly: Semi-supervised anomaly detection via adversarial training. In *Asian conference on computer vision*. Springer, 622–637.

[4] Samet Akçay, Amir Atapour-Abarghouei, and Toby P Breckon. 2019. Skip-ganomaly: Skip connected and adversarially trained encoder-decoder anomaly detection. In *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–8.

[5] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. 2018. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 139–154.

[6] Simon Duque Anton, Suneetha Kanoor, Daniel Fraunholz, and Hans Dieter Schotten. 2018. Evaluation of machine learning-based anomaly detection algorithms on an industrial modbus/tcp data set. In *Proceedings of the 13th international conference on availability, reliability and security*. 1–9.

[7] Pierre Baldi. 2012. Autoencoders, unsupervised learning, and deep architectures. In *Proceedings of ICML workshop on unsupervised and transfer learning*. JMLR Workshop and Conference Proceedings, 37–49.

[8] Patrick Billingsley. 2008. *Probability and measure*. John Wiley & Sons.

[9] Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K Warmuth. 1987. Occam's razor. *Information processing letters* 24, 6 (1987), 377–380.

[10] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. 2000. LOF: identifying density-based local outliers. In *SIGMOD*. 93–104.

[11] Richard L Burden, J Douglas Faires, and Annette M Burden. 2015. *Numerical analysis*. Cengage learning.

[12] Yanbei Chen, Xiatian Zhu, and Shaogang Gong. 2018. Semi-supervised deep learning with memory. In *Proceedings of the European conference on computer vision (ECCV)*. 268–283.

[13] Frederik Michel Dekking, Cornelis Kraaikamp, Hendrik Paul Lopuhaä, and Ludolf Erwin Meester. 2005. *A Modern Introduction to Probability and Statistics: Understanding why and how*. Vol. 488. Springer.

[14] Min Du, Zhi Chen, Chang Liu, Rajvardhan Oak, and Dawn Song. 2019. Lifelong anomaly detection through unlearning. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. 1283–1297.

[15] Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter. 2019. Neural architecture search: A survey. *The Journal of Machine Learning Research* 20, 1 (2019), 1997–2017.

[16] Andrew Emmott, Shubhomoy Das, Thomas Dietterich, Alan Fern, and Weng-Keen Wong. 2015. A meta-analysis of the anomaly detection problem. *ArXiv* 1503.01158 (2015). https://arxiv.org/abs/1503.01158

[17] Eduardo Dadalto Camara Gomes, Florence Alberge, Pierre Duhamel, and Pablo Piantanida. 2021. Igeood: An Information Geometry Approach to Out-of-Distribution Detection. In *International Conference on Learning Representations*.

[18] Parikshit Gopalan, Vatsal Sharan, and Udi Wieder. 2019. Pidforest: anomaly detection via partial identification. *Advances in Neural Information Processing Systems* 32 (2019).

[19] Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. 2021. Revisiting deep learning models for tabular data. *Advances in Neural Information Processing Systems* 34 (2021), 18932–18943.

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

[21] Yves Grandvalet and Yoshua Bengio. 2004. Semi-supervised learning by entropy minimization. *Advances in neural information processing systems* 17 (2004).

[22] Songqiao Han, Xiyang Hu, Hailiang Huang, Minqi Jiang, and Yue Zhao. [n.d.]. ADBench: Anomaly Detection Benchmark. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

[23] Zengyou He, Xiaofei Xu, and Shengchun Deng. 2003. Discovering cluster-based local outliers. *Pattern recognition letters* 24, 9-10 (2003), 1641–1650.

[24] Waleed Hilal, S Andrew Gadsden, and John Yawney. 2021. A review of anomaly detection techniques and applications in financial fraud. *Expert Systems with Applications* (2021), 116429.

[25] Zongyuan Huang, Baohua Zhang, Guoqiang Hu, Longyuan Li, Yanyan Xu, and Yaohui Jin. 2021. Enhancing Unsupervised Anomaly Detection with Score-Guided Network. *arXiv preprint arXiv:2109.04684* (2021).

[26] Dihong Jiang, Sun Sun, and Yaoliang Yu. 2021. Revisiting flow generative models for Out-of-distribution detection. In *International Conference on Learning Representations*.

[27] Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. 2021. Reversible Instance Normalization for Accurate Time-Series Forecasting against Distribution Shift. In *International Conference on Learning Representations*.

[28] Diederik P. Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.).

[29] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences* 114, 13 (2017), 3521–3526.

[30] Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images. (2009).

[31] Aleksandar Lazarevic, Levent Ertoz, Vipin Kumar, Aysel Ozgur, and Jaideep Srivastava. 2003. A comparative study of anomaly detection schemes in network intrusion detection. In *SDM*. SIAM, 25–36.

[32] Dongha Lee, Sehun Yu, Hyunjun Ju, and Hwanjo Yu. 2021. Weakly supervised temporal anomaly segmentation with dynamic time warping. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 7355–7364.

[33] Meng-Chieh Lee, Shubhranshu Shekhar, Christos Faloutsos, T Noah Hutson, and Leon Iasemidis. 2021. Gen 2 Out: Detecting and Ranking Generalized Anomalies. In *Big Data*. IEEE, 801–811.

[34] Yuening Li, Zhengzhang Chen, Daochen Zha, Kaixiong Zhou, Haifeng Jin, Haifeng Chen, and Xia Hu. 2021. Autood: Neural architecture search for outlier detection. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*. IEEE, 2117–2122.

[35] Zhe Li, Chunhua Sun, Chunli Liu, Xiayu Chen, Meng Wang, and Yezheng Liu. 2022. Dual-MGAN: An Efficient Approach for Semi-supervised Outlier Detection with Few Identified Anomalies. *ACM Transactions on Knowledge Discovery from Data (TKDD)* (2022).

[36] Zheng Li, Yue Zhao, Xiyang Hu, Nicola Botta, Cezar Ionescu, and George Chen. 2022. Ecod: Unsupervised outlier detection using empirical cumulative distribution functions. *IEEE Transactions on Knowledge and Data Engineering* (2022).

[37] Boyang Liu, Pang-Ning Tan, and Jiayu Zhou. 2022. Unsupervised Anomaly Detection by Robust Density Estimation. In *AAAI*. AAAI Press, 4101–4108. https://ojs.aaai.org/index.php/AAAI/article/view/20328

[38] Chenxi Liu, Barret Zoph, Maxim Neumann, Jonathon Shlens, Wei Hua, Li-Jia Li, Li Fei-Fei, Alan Yuille, Jonathan Huang, and Kevin Murphy. 2018. Progressive neural architecture search. In *Proceedings of the European conference on computer vision (ECCV)*. 19–34.

[39] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. 2008. Isolation forest. In *2008 eighth ieee international conference on data mining*. IEEE, 413–422.

[40] Siyan Liu, Pei Zhang, Dan Lu, and Guannan Zhang. 2021. PI3NN: Out-of-distribution-aware Prediction Intervals from Three Neural Networks. In *International Conference on Learning Representations*.

[41] Yezheng Liu, Zhe Li, Chong Zhou, Yuanchun Jiang, Jianshan Sun, Meng Wang, and Xiangnan He. 2019. Generative adversarial active learning for unsupervised outlier detection. *IEEE Transactions on Knowledge and Data Engineering* (2019).

[42] Clare Lyle, Lisa Schut, Robin Ru, Yarin Gal, and Mark van der Wilk. 2020. A bayesian perspective on training speed and model selection. *Advances in Neural Information Processing Systems* 33 (2020), 10396–10408.

[43] Jitendra Singh Malik, Guansong Pang, and Anton van den Hengel. 2022. Deep Learning for Hate Speech Detection: A Comparative Study. *arXiv preprint arXiv:2202.09517* (2022).

[44] Rafael Martinez-Guerra and Juan Luis Mata-Machuca. 2016. *Fault detection and diagnosis in nonlinear systems*.

[45] Glenn W Milligan. 1985. An algorithm for generating artificial test clusters. *Psychometrika* 50, 1 (1985), 123–127.

[46] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. 2011. Reading digits in natural images with unsupervised feature learning. (2011).

[47] Salima Omar, Asri Ngadi, and Hamid H Jebur. 2013. Machine learning techniques for anomaly detection: an overview. *International Journal of Computer Applications* 79, 2 (2013).

[48] Guansong Pang, Longbing Cao, Ling Chen, and Huan Liu. 2018. Learning representations of ultrahigh-dimensional data for random distance-based outlier detection. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 2041–2050.

[49] 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).

[50] Guansong Pang, Chunhua Shen, Huidong Jin, and Anton van den Hengel. 2019. Deep weakly-supervised anomaly detection. *arXiv preprint arXiv:1910.13601* (2019).

[51] Guansong Pang, Chunhua Shen, and Anton van den Hengel. 2019. Deep anomaly detection with deviation networks. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 353–362.

[52] Emilio Parisotto, Lei Jimmy Ba, and Ruslan Salakhutdinov. 2016. Actor-Mimic: Deep Multitask and Transfer Reinforcement Learning. In *ICLR (Poster)*.

[53] Tahereh Pourhabibi, Kok-Leong Ong, Booi H Kam, and Yee Ling Boo. 2020. Fraud detection: A systematic literature review of graph-based anomaly detection approaches. *Decision Support Systems* 133 (2020), 113303.

[54] Theodoros Rekatsinas, Saurav Ghosh, Sumiko R Mekaru, Elaine O Nsoesie, John S Brownstein, Lise Getoor, and Naren Ramakrishnan. 2015. Sourceseer: Forecasting rare disease outbreaks using multiple data sources. In *Proceedings of the 2015 SIAM International Conference on Data Mining*. SIAM, 379–387.

[55] Douglas A Reynolds. 2009. Gaussian mixture models. *Encyclopedia of biometrics* 741, 659-663 (2009).

[56] Oren Rippel, Manohar Paluri, Piotr Dollar, and Lubomir Bourdev. 2015. Metric learning with adaptive density discrimination. *arXiv preprint arXiv:1511.05939* (2015).

[57] Lukas Ruff, Jacob R Kauffmann, Robert A Vandermeulen, Grégoire Montavon, Wojciech Samek, Marius Kloft, Thomas G Dietterich, and Klaus-Robert Müller. 2021. A unifying review of deep and shallow anomaly detection. *Proc. IEEE* 109, 5 (2021), 756–795.

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

[59] Lukas Ruff, Robert A. Vandermeulen, Nico Görnitz, Alexander Binder, Emmanuel Müller, Klaus-Robert Müller, and Marius Kloft. 2020. Deep Semi-Supervised Anomaly Detection. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

[60] Bernhard Schölkopf, Robert C Williamson, Alexander J Smola, John Shawe-Taylor, John C Platt, et al. 1999. Support vector method for novelty detection.. In *NIPS*, Vol. 12. Citeseer, 582–588.

[61] Jonas Soenen, Elia Van Wolputte, Lorenzo Perini, Vincent Vercruyssen, Wannes Meert, Jesse Davis, and Hendrik Blockeel. 2021. The Effect of Hyperparameter Tuning on the Comparative Evaluation of Unsupervised Anomaly Detection Methods. In *Proceedings of the KDD'21 Workshop on Outlier Detection and Description*. Outlier Detection and Description Organising Committee, 1–9.

[62] Georg Steinbuss and Klemens Böhm. 2021. Benchmarking unsupervised outlier detection with realistic synthetic data. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 15, 4 (2021), 1–20.

[63] Waqas Sultani, Chen Chen, and Mubarak Shah. 2018. Real-world anomaly detection in surveillance videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 6479–6488.

[64] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).

[65] Xinshao Wang, Yang Hua, Elyor Kodirov, Guosheng Hu, Romain Garnier, and Neil M Robertson. 2019. Ranked list loss for deep metric learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 5207–5216.

[66] Robert F Woolson. 2007. Wilcoxon signed-rank test. *Wiley encyclopedia of clinical trials* (2007), 1–3.

[67] Jiaxuan You, Zhitao Ying, and Jure Leskovec. 2020. Design space for graph neural networks. *Advances in Neural Information Processing Systems* 33 (2020), 17009–17021.

[68] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. 2015. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365* (2015).

[69] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. 2020. Gradient surgery for multi-task learning. *Advances in Neural Information Processing Systems* 33 (2020), 5824–5836.

[70] Weiren Yu, Jianxin Li, Md Zakirul Alam Bhuiyan, Richong Zhang, and Jinpeng Huai. 2017. Ring: Real-time emerging anomaly monitoring system over text streams. *IEEE Transactions on Big Data* 5, 4 (2017), 506–519.

[71] Houssam Zenati, Chuan Sheng Foo, Bruno Lecouat, Gaurav Manek, and Vijay Ramaseshan Chandrasekhar. 2018. Efficient gan-based anomaly detection. *arXiv preprint arXiv:1802.06222* (2018).

[72] Houssam Zenati, Manon Romain, Chuan-Sheng Foo, Bruno Lecouat, and Vijay Chandrasekhar. 2018. Adversarially learned anomaly detection. In *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, 727–736.

[73] Friedemann Zenke, Ben Poole, and Surya Ganguli. 2017. Continual learning through synaptic intelligence. In *International Conference on Machine Learning*. PMLR, 3987–3995.

[74] Liangwei Zhang, Jing Lin, and Ramin Karim. 2018. Adaptive kernel density-based anomaly detection for nonlinear systems. *Knowledge-Based Systems* 139 (2018), 50–63.

[75] Jun Zhao, Xudong Liu, Qiben Yan, Bo Li, Minglai Shao, and Hao Peng. 2020. Multi-attributed heterogeneous graph convolutional network for bot detection. *Information Sciences* 537 (2020), 380–393.

[76] Yue Zhao, Xiyang Hu, Cheng Cheng, Cong Wang, Changlin Wan, Wen Wang, Jianing Yang, Haoping Bai, Zheng Li, Cao Xiao, et al. 2021. SUOD: Accelerating large-scale unsupervised heterogeneous outlier detection. *MLSys* 3 (2021), 463–478.

[77] Yue Zhao, Zain Nasrullah, and Zheng Li. 2019. PyOD: A Python Toolbox for Scalable Outlier Detection. *Journal of Machine Learning Research* 20 (2019), 1–7.

[78] Yue Zhao, Guoqing Zheng, Subhabrata Mukherjee, Robert McCann, and Ahmed Awadallah. 2022. Admoe: Anomaly detection with mixture-of-experts from noisy labels. *arXiv preprint arXiv:2208.11290* (2022).

[79] Yingjie Zhou, Xucheng Song, Yanru Zhang, Fanxing Liu, Ce Zhu, and Lingqiao Liu. 2021. Feature encoding with autoencoders for weakly supervised anomaly detection. *IEEE Transactions on Neural Networks and Learning Systems* (2021).

[80] Zhi-Hua Zhou. 2018. A brief introduction to weakly supervised learning. *National science review* 5, 1 (2018), 44–53.

[81] 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 *International conference on learning representations*.