Unsupervised Anomaly Detection for Tabular Data Using Deep Noise Evaluation

Wei Dai, Kai Hwang, Jicong Fan*

The Chinese University of Hong Kong, Shenzhen, China
weidai@link.cuhk.edu.com, hwangkai@cuhk.edu.cn, fanjicong@cuhk.edu.cn

Abstract

Unsupervised anomaly detection (UAD) plays an important role in modern data analytics and it is crucial to provide simple yet effective and guaranteed UAD algorithms for real applications. In this paper, we present a novel UAD method for tabular data by evaluating how much noise is in the data. Specifically, we propose to learn a deep neural network from the clean (normal) training dataset and a noisy dataset, where the latter is generated by adding highly diverse noises to the clean data. The neural network can learn a reliable decision boundary between normal data and anomalous data when the diversity of the generated noisy data is sufficiently high so that the hard abnormal samples lie in the noisy region. Importantly, we provide theoretical guarantees, proving that the proposed method can detect anomalous data successfully, although the method does not utilize any real anomalous data in the training stage. Extensive experiments through more than 60 benchmark datasets demonstrate the effectiveness of the proposed method in comparison to 12 baselines of UAD. Our method obtains a 92.27% AUC score and a 1.68 ranking score on average. Moreover, compared to the state-of-the-art UAD methods, our method is easier to implement.

Introduction

In the realm of data analysis, anomaly detection (AD) stands as a pivotal challenge with far-reaching implications across various domains, including cybersecurity (Siddiqui et al. 2019; Saeed et al. 2023), healthcare (Yang, Qi, and Zhou 2023; Abououf et al. 2023), finance (Hilal, Gadsden, and Yawney 2022), and industrial processes (Fan, Chow, and Qin 2022; Roth et al. 2022). Existing deep learning-based unsupervised AD methods often rely on an auxiliary learning objective such as auto-encoder, generative model, and contrastive learning. These methods indirectly detect anomalous data using other metrics such as reconstruction error, which lack generalizability and reliability guarantees (Hussain et al. 2023). Explicitly learning a one-class decision boundary may resolve this issue. Many well-known unsupervised AD methods assume the normal training data has a special structure in their data space or embedding space (Schölkopf et al. 2001; Tax and Duin 2004; Ruff et al. 2018; Goyal et al. 2020; Zhang et al. 2024; Xiao et al. 2025). Such assumptions may not hold or be guaranteed in practice and sometimes place a burden on the model training (Cai and Fan 2022; Fu, Zhang, and Fan 2024). For instance, in deep SVDD (Ruff et al. 2018), the optimal decision boundary in the embedding space may be very different from the learned hypersphere, leading to unsatisfactory detection performance (Zhang et al. 2024).

Given that tabular data is probably the most common data type and other types of data such as images can be converted to tabular data using feature encoders or pretrained models, in this work, we focus on the tabular data only. We propose a novel unsupervised AD method for tabular data without making any assumption about the distribution of normal data. Since hard anomalies are often close to normal data, it is reasonable to hypothesize that hard anomalies are special cases of perturbed samples of normal data. Therefore, if the diversity of the perturbations or added noises on normal data is sufficiently high, we can obtain hard anomalies. Consequently, if a model can recognize highly diverse perturbations or noises, it can detect hard anomalies as well as easy anomalies successfully. By directly learning from the diverse noise patterns and the clean data patterns, we can learn an effective decision boundary around the normal data, generalizing well to unseen data. Our contributions are highlighted as follows.

- We propose a novel AD method for tabular data using noise evaluation. Our scheme generates highly diverse noise-augmented instances for the normal samples. By evaluating the noise magnitude, our method can accurately identify anomalies.
- The proposed method provides a simple yet effective scheme that does not make assumptions about the normal training data. In addition, the noise generation is straightforward without any extra training. Compared with (Wang et al. 2021; Goyal et al. 2020; Yan et al. 2021; Cai and Fan 2022), our method is more lightweight for training (requires less module for training.)
- We theoretically prove the generalizability and reliability of the proposed method.
- We conduct extensive empirical experiments on 47 real datasets in an unsupervised anomaly detection setting and 25 real-world tabular datasets in a one-class classification setting to demonstrate the performance of the proposed schemes. The results show that our method achieved su-

*Corresponding author.

perior performance compared with 12 baseline methods including the state-of-the-art.

Related Work

Unsupervised anomaly detection (UAD) is also known as the one-class classification (OCC) problem, in which all or most training samples are assumed to be normal. The learning objective is to learn a decision boundary that distinguishes whether a sample belongs to the same distribution of the normal training data or not. There is another similar problem setting known as outlier detection on contaminated dataset (Huang et al. 2024). The goal is to detect noised samples or outliers within the training data (Ding, Zhao, and Akoglu 2022). This line of work is orthogonal to our UAD problem.

In the past decades, many UAD methods have been studied (Liu, Ting, and Zhou 2008; Chang et al. 2023). Traditional methods like proximity-based (Breunig et al. 2000; Angiulli and Pizzuti 2002; Papadimitriou et al. 2003; He, Xu, and Deng 2003), probabilty-based (Yang, Latecki, and Pokrajac 2009; Zong et al. 2018; Li et al. 2020), and one-class support vector machine (Schölkopf et al. 2001; Tax and Duin 2004) approaches struggle with high dimensionality and complex data structures. Deep neural network-based methods have been proposed to address these issues. For instance, auto-encoder methods identify outliers by detecting high reconstruction errors, as outlier samples do not conform to historical data patterns (Aggarwal 2016; Chen et al. 2017; Wang et al. 2021). Generative model methods compare latent features or generated samples to spot anomalies (Schlegl et al. 2017; Liu et al. 2019; Zhang et al. 2023; Tur et al. 2023; Xiao et al. 2025). For example, (Xiao et al. 2025) proposed an inverse generative adversarial network that converts the data distribution to a compact Gaussian distribution, based on which the density of test data can be calculated for anomaly detection. Contrastive learning (Sohn et al. 2020; Jin et al. 2021; Shenkar and Wolf 2022) leverages feature representation differences to detect anomalies. Unlike autoencoder-based methods that focus on reducing reconstruction error and reducing dimensionality to remove noise, our method aims to evaluate noise level, which is similar to the denoising diffusion model (Ho, Jain, and Abbeel 2020).

Some works explicitly build an anomaly detection or OCC objective (Ruff et al. 2018; Goyal et al. 2020; Yan et al. 2021; Chen et al. 2022; Cai and Fan 2022). For instance, Deep SVDD (Ruff et al. 2018) trains a neural network to construct a hypersphere in the output space to enclose the normal training data. DROCC (Goyal et al. 2020) assumes normal samples lie on low-dimensional manifolds and treats identifying the ideal hypersphere as an adversarial optimization problem. PLAD (Cai and Fan 2022) outputs an anomaly score by learning a small perturbation of normal data as the negative sample with a classifier. Unlike PLAD, which uses extra additive and multiplicative perturbations requiring a perturbator, our method generates negative samples without extra parameters, making the training efficient.

It is worth mentioning that there are vision UAD methods utilizing synthetic anomalous data. However, the normality and abnormality can be visualized directly and there naturally exists prior knowledge about anomalies (e.g., visible spots or blurs) in visual data. Regarding tabular data, we do not have such prior knowledge. Vision AD methods require prior pre-trained models or external reference datasets to obtain the negative samples, such as DREAM (Zavrtanik, Kristan, and Skočaj 2021) with the Describable Textures Dataset or AnomalyDiffusion (Hu et al. 2024) with a stable diffusion model. Such resources are costly and violate the principle of unsupervised learning. Tabular data, however, spans diverse domains (e.g., medical, industrial), making external datasets and pretrained model infeasible.

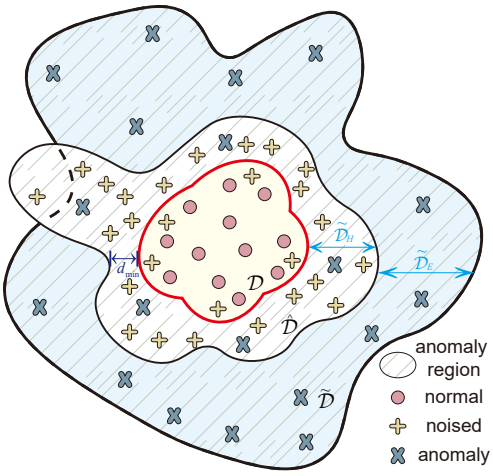


Figure 1: An illustration of the allocation of normal, noised, and true anomalous samples. $\mathcal{D}$, $\hat{\mathcal{D}}$, and $\tilde{\mathcal{D}}$ are the normal, noised, and anomalous distributions respectively. $\tilde{\mathcal{D}}$ is composed of a hard part $\tilde{\mathcal{D}}_H$ and an easy part $\tilde{\mathcal{D}}_E$. Theorem 1 and Theorem 2 are for $\tilde{\mathcal{D}}_H$ and $\tilde{\mathcal{D}}_E$ respectively.

Symbol	Description	Symbol	Description
x	a scalar	$\boldsymbol{x}_i$	a vector with index i
$\mathcal{X}_i$	a set with index i	$[i]$	the set $\{1, 2, \ldots, i\}$
$\mathcal{D}$	a distribution	$\|\cdot\|_2$	ℓ_2 norm of vector
$\|\cdot\|_1$	ℓ_1 norm of vector	$\|\boldsymbol{x}\|$	element-wise absolute
$\cup$	union of sets	$\subset$	subset
H	entropy	$h_{\boldsymbol{\theta}}$	model parameterized by $\boldsymbol{\theta}$

Table 1: Notations

Proposed Method

Problem Formulation and Notations

Given a data set $\mathcal{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$, where the element $\boldsymbol{x}_i$ are drawn from an unknown distribution $\mathcal{D}$ in $\mathbb{R}^d$ (deemed as a normal data distribution). In the context of UAD, the primary objective is to develop a function $f : \mathbb{R}^d \to \{0, 1\}$, which effectively discriminates between in-distribution (normal) and out-of-distribution (anomalous) instances. This discriminative function is formulated to assign a binary label, where $f(x) = 1$ indicates $\boldsymbol{x}$ does not belong to $\mathcal{D}$ and $f(x) = 0$ corresponds to $\boldsymbol{x}$ coming from $\mathcal{D}$. The main notations used in this paper are shown in Table 1.

Anomalous Data Decomposition

Let $\tilde{\mathcal{X}}$ be the set consisting of all anomalous data drawn from some unknown distribution $\tilde{\mathcal{D}}$ deemed as an anomalous distribution. For any $\tilde{\boldsymbol{x}} \in \tilde{\mathcal{X}}$, we decompose it as

$$\tilde{\boldsymbol{x}} = \boldsymbol{x} + \boldsymbol{\epsilon}, \tag{1}$$

where $\boldsymbol{x} \in \mathcal{D}$ is a normal counterpart of $\tilde{\boldsymbol{x}}$ and $\boldsymbol{\epsilon}$ denotes the derivation of $\tilde{\boldsymbol{x}}$ from $\boldsymbol{x}$. The magnitude of $\boldsymbol{\epsilon}$, denoted as $\|\boldsymbol{\epsilon}\|_1$, measures how anomalous $\tilde{\boldsymbol{x}}$ is. Note that this decomposition is not unique and hence one may seek the one with the smallest $\|\boldsymbol{\epsilon}\|_1$. If we can learn a model h to predict $\boldsymbol{\epsilon}$ for $\tilde{\boldsymbol{x}}$, i.e.,

$$\boldsymbol{\epsilon} = h(\tilde{\boldsymbol{x}}), \tag{2}$$

we will be able to determine whether $\tilde{\boldsymbol{x}}$ is normal or not according to $\|\boldsymbol{\epsilon}\|_1$. The challenge is that there is no available information about $\tilde{\mathcal{X}}$ in the training stage and we can only utilize $\mathcal{X}$.

Although $\tilde{\mathcal{X}}$ is unknown, we further theoretically partition $\tilde{\mathcal{X}}$ into two subsets without overlap, i.e.,

$$\tilde{\mathcal{X}} \triangleq \tilde{\mathcal{X}}_E \cup \tilde{\mathcal{X}}_H,\ \tilde{\mathcal{X}}_E \cap \tilde{\mathcal{X}}_H = \emptyset, \tilde{\mathcal{X}}_E \sim \tilde{\mathcal{D}}_E, \tilde{\mathcal{X}}_H \sim \tilde{\mathcal{D}}_H. \tag{3}$$

$\tilde{\mathcal{X}}_E$ denotes an easy set, drawn from the easy part $\tilde{\mathcal{D}}_E$ of $\tilde{\mathcal{D}}$, in which $\|\boldsymbol{\epsilon}\|_1$ for each sample is sufficiently large, while $\tilde{\mathcal{X}}_H$ denotes a hard set, drawn from the hard part $\tilde{\mathcal{D}}_H$ of $\tilde{\mathcal{D}}$, in which $\|\boldsymbol{\epsilon}\|_1$ for each sample is small. After the partition, we can assert that $\tilde{\mathcal{X}}_H$ is closer to the normal data. Consequently, it is easier for a model to recognize samples in $\tilde{\mathcal{X}}_E$ than those in $\tilde{\mathcal{X}}_H$, as the $\|\boldsymbol{\epsilon}\|_1$ values of samples in $\tilde{\mathcal{D}}_E$ are significantly larger than those in $\tilde{\mathcal{D}}_H$. Figure 1 provides an intuitive example. The ultimate goal is to learn the decision boundary around the normal data.

Here, we focus on how to detect the samples in $\tilde{\mathcal{X}}_H$ or drawn from $\tilde{\mathcal{D}}_H$. Since $\tilde{\mathcal{X}}_H$ is very close to the normal data, it is reasonable to hypothesize that hard anomalies are special cases of perturbed samples of normal data. We propose to generate a noisy dataset $\hat{\mathcal{X}} \subset \mathbb{R}^d$ from $\mathcal{X}$ by adding various noise to $\mathcal{X}$, and assume $\hat{\mathcal{X}}$ is drawn from certain perturbed distribution $\hat{\mathcal{D}}$, i.e.,

$$\hat{\mathcal{X}} \leftarrow \text{Gen}(\mathcal{X}) \sim \hat{\mathcal{D}}, \tag{4}$$

where Gen denotes the noisy data generator and $|\hat{\mathcal{X}}| \gg N$. Let the diversity of added noise is sufficiently large, such that

$$\tilde{\mathcal{X}}_H \subset \hat{\mathcal{X}}, \tag{5}$$

i.e., anomaly patterns of the hard set $\tilde{\mathcal{X}}_H$ are included in $\hat{\mathcal{X}}$. Even if (5) does not hold, as shown by Theorem 1, it is still possible to obtain correct detection, provided that $\hat{\mathcal{D}}$ is not too far from $\tilde{\mathcal{D}}_H$, i.e.,

$$\text{dist}(\hat{\mathcal{D}}, \tilde{\mathcal{D}}_H) \leq \gamma, \tag{6}$$

where $\text{dist}(\cdot,\cdot)$ is some distance or divergence measure between two distributions and γ is not too large. Therefore, a model h learned from $\hat{\mathcal{X}}$ is able to generalize to $\tilde{\mathcal{X}}_H$ and then detect anomaly.

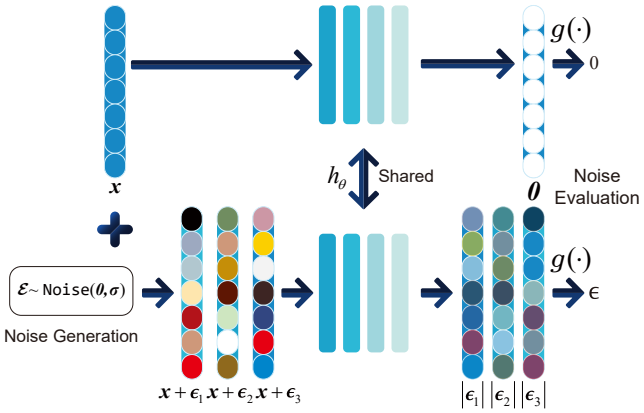


Figure 2: The training process of noise evaluation model. Noise with 0 mean and σ standard deviation is added to the original data $\boldsymbol{x}$ to create noised versions $\hat{\boldsymbol{x}} = \boldsymbol{x} + \boldsymbol{\epsilon}$. The model $h_{\boldsymbol{\theta}}$ is trained to discern the zero vector for the original data and identify the noise vector $|\boldsymbol{\epsilon}|$ for the noised data. The final anomaly decision is made using an aggregation function $g(\cdot)$, where high-magnitude noise indicates abnormality.

Noise Evaluation Model

To generate $\hat{\mathcal{X}}$, we add random noise to the elements of each sample $\boldsymbol{x} \in \mathcal{X}$. This operation will reduce the quality of the data, that is, the quality of $\hat{\mathcal{X}}$ is lower than that of $\mathcal{X}$, supported by

Proposition 1. Adding random noises independently to the entries of $\mathcal{X}$ makes the data more disordered (higher entropy).

We would like to learn a deep neural network $h(\cdot) : \mathbb{R}^d \to \mathbb{R}^d$ parameterized by $\boldsymbol{\theta}$ to quantify the quality of the input data. Its output is a vector with the same size as the input while each entry tells whether the input feature is noised or not. We generate the noised dataset by

$$\hat{\mathcal{X}} = \hat{\mathcal{X}}_1 \cup \hat{\mathcal{X}}_2 \cdots \cup \hat{\mathcal{X}}_K, \tag{7}$$

where each $\hat{\mathcal{X}}_k$ is composed of the samples generated by

$$\hat{\boldsymbol{x}} = \boldsymbol{x} + \boldsymbol{\epsilon},\ \ \boldsymbol{\epsilon} \sim \text{Noise}_k(\mathbf{0}, \boldsymbol{\sigma}_k),\ \ \boldsymbol{x} \in \mathcal{X}. \tag{8}$$

In (8), $\text{Noise}_k(\mathbf{0}, \boldsymbol{\sigma}_k)$ (to be detailed in the next section) is a multivariate noise distribution with $\mathbf{0}$ mean value and $\boldsymbol{\sigma}_k$ standard deviation in $\mathbb{R}^d$, abbreviated as $\text{Noise}(\boldsymbol{\sigma}_k)$. For instance, Noise_k can be a Gaussian distribution. We also denote the standard deviation of the noise as *noise level* [1]. According to (7) and (8), we see that $\hat{\mathcal{X}}$ can contain different types of noise distributions with different standard deviations, and the diversity is controlled by $\sum_{k=1}^{K} \|\sigma_k\|$. Refer to Table 2 for the choices of the noise generator. Then, we write the learning objective as

$$\min_{\boldsymbol{\theta}} \sum_{\boldsymbol{x}_i \in \mathcal{X}} \|h_{\boldsymbol{\theta}}(\boldsymbol{x}_i) - \mathbf{0}\|_2^2 + \sum_{\hat{\boldsymbol{x}}_i \in \hat{\mathcal{X}}} \|h_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}_i) - |\boldsymbol{\epsilon}_i|\|_2^2, \tag{9}$$

where $\mathbf{0}$ is a d-dimension vector with all zero values, $\boldsymbol{\epsilon}$ is drawn from some $\text{Noise}(0, \boldsymbol{\sigma})$, and $\hat{\boldsymbol{x}} = \boldsymbol{x} + \boldsymbol{\epsilon}$ for some

[1] We use noise level and standard deviation of noise interchangeably in this paper.

$\boldsymbol{x} \in \mathcal{X}$. Note that in (9), there is no hyperparameter to determine except the network structure, while many previous methods such as (Ruff et al. 2018; Goyal et al. 2020; Cai and Fan 2022; Zhang et al. 2024) have at least one more crucial hyperparameter to tune in the learning objective.

In (9), the absolute value is used to let the model output a positive signal indicating an anomaly. Here, we only require the model to predict how much the noise is instead of the exact noise value, which also reduces the difficulty of the learning process. In addition, our method is closely related to the denoising score matching (Song and Ermon 2019; Vincent 2011), which estimates the gradient of the probability density of the data distribution. We prove (9) is a lower bound of denoising score matching learning objective in Appendix A. Hence, our method also learns data distribution $\mathcal{D}$ by proxy. Predicting the noise magnitude has the following advantages.

Claim 1. Estimating the element-wise magnitude of noise enables the model to quantify the deviation of a sample from the normal data distribution $\mathcal{D}$. It improves the ability to identify specific features with abnormalities.

Since the model output is a d-dimension vector, we introduce an aggregation operation after the training, $g(\cdot)$, to map the output vector to a scalar for anomaly detection. The operation in our design can be maximum, minimum, mean, median, or a combination. The final anomaly score is determined by

$$\text{score}(\boldsymbol{x}) := g\left(h_{\boldsymbol{\theta}}(\boldsymbol{x})\right). \tag{10}$$

Taking the maximum as an example, we have $\text{score}(\boldsymbol{x}) = \max_{i\in[d]}[h_{\boldsymbol{\theta}}(\boldsymbol{x})]_i$. We need to determine a threshold $\tau > 0$ for the anomaly score. If $\text{score}(\boldsymbol{x}) > \tau$, $\boldsymbol{x}$ is anomalous. An overview of our model is shown in Figure 2.

Noise Generation and Model Training

The noise generation in our design is not arbitrary. To cover the sampling space of the noised distributions as much as possible, we randomly generate the noise for each training sample in terms of noise level and position. An intuitive example is as follows:

$$\boldsymbol{\epsilon} = \begin{bmatrix} \overbrace{}^{\sim\text{Noise}(\sigma_2)} & & \overbrace{\phantom{\epsilon_{i+2}}}^{\sim\text{Noise}(\sigma_1)} & & \overbrace{\phantom{\epsilon_{d-1},\epsilon_d}}^{\sim\text{Noise}(\sigma_1)} \\ \underbrace{\epsilon_1,\epsilon_2}_{\sim\text{Noise}(\sigma_1)},\overbrace{\epsilon_3,\epsilon_4}, & ..., & \underbrace{\epsilon_i,\epsilon_{i+1}}_{\sim\text{Noise}(\sigma_3)},\overbrace{\epsilon_{i+2}}, & ..., & \underbrace{\epsilon_{d-2}}_{\sim\text{Noise}(\sigma_m)},\overbrace{\epsilon_{d-1},\epsilon_d} \end{bmatrix}. \tag{11}$$

For one type of distribution (e.g. Gaussian), there are often two hyper-parameters to control the noise generation process. The first one is σ_{max}, denoting the maximum noise level. The other is m, describing how many parts of the feature vector are added by the same level of noise. Therefore, for an $\boldsymbol{x}$, different elements may be corrupted by different levels of noise, and, if necessary, one $\boldsymbol{x}$ can produce multiple $\hat{\boldsymbol{x}}$ with different types of distribution. A detailed process for generating noised samples on a batch of data is in Algorithm 1. According to Algorithm 1, the noise generation time complexity is $\mathcal{O}(bd)$. In contrast, other methods involving perturbation (Cai and Fan 2022; Qiu et al. 2021) and adversarial sample (Goyal et al. 2020) have time complexity with $\mathcal{O}(bdW)$, where W is workload related to a neural network module. Compared with them, our noise generation is more efficient, where no learnable parameter is required. A comparative study on time cost is shown in Appendix I.

Algorithm 1: Noise Generation

Input: maximum noise level σ_{max}, number of noise distributions m, batch size b, the dimensionality of data d.

$\Delta \leftarrow [0, \frac{1}{m}\sigma_{max}, \frac{2}{m}\sigma_{max}, ..., \frac{m}{m}\sigma_{max}]$ ▷ make m intervals
Initialize $\boldsymbol{\mathcal{E}}$ with empty
for $j \in \{1, ..., b\}$ **do**
  for $i \in \{1, ..., m\}$ **do**
    //random noise level
    $\hat{\sigma} \leftarrow \texttt{Uniform}(\Delta[i], \Delta[i+1])$
    $\boldsymbol{\epsilon}[\frac{i-1}{m}d : \frac{i}{m}d] \leftarrow \texttt{Noise}(0, \hat{\sigma})$ *//generate noise from m noise levels for m parts*
  end for
  //shuffle position of noise elements
  $\boldsymbol{\epsilon} \leftarrow \texttt{Shuffle}(\boldsymbol{\epsilon})$
  $\boldsymbol{\mathcal{E}}[j] \leftarrow \boldsymbol{\epsilon}$
end for

Output: Generated noise $\boldsymbol{\mathcal{E}}$ for a batch of input data

In Figure 2, the lower pathway illustrates the noise synthesis mechanism, where a noise vector $\boldsymbol{\epsilon}$ (mean 0, standard deviation $\boldsymbol{\sigma}$) is randomly generated and added to the input $\boldsymbol{x}$, producing a noise-augmented variant $\hat{\boldsymbol{x}} = \boldsymbol{x} + \boldsymbol{\epsilon}$. Multiple noised samples can be generated from a single input. Both $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$ are processed by the noise evaluation network h_θ, optimized to regress towards zero for $\boldsymbol{x}$ and estimate the noise vector $|\boldsymbol{\epsilon}|$ for $\hat{\boldsymbol{x}}$. Training details are in Algorithm 2 of Appendix B. Notice that for each training epoch, we randomly generate a new noised instance for each training sample. This helps us enlarge the sampling number from the noise distribution, and avoid over-fitting on some ineffective noise. There are some optional noise generation schemes such as using different noise types, different noise levels, and different noise ratios, which are studied later. These options add several hyper-parameters to our methods. However, it is still simpler than many recent methods. We compare them in Appendix J.

Theoretical Analysis

In the proposed method, we learn a one-class classification decision boundary closely around the normal samples. In Figure 1, we divide the anomaly region into two non-overlap distributions, easy $\tilde{\mathcal{D}}_E$, and hard $\tilde{\mathcal{D}}_H$, respectively. In this section, we theoretically show the anomaly detection ability of the model meeting easy anomaly samples $\tilde{\mathcal{X}}_E$ from $\tilde{\mathcal{D}}_E$ and hard anomaly samples $\tilde{\mathcal{X}}_H$ from $\tilde{\mathcal{D}}_H$ in Theorem 1 and Theorem 2, respectively.

Without loss of generality, we let $h_{\boldsymbol{\theta}} \in \mathcal{H}$, where $\mathcal{H}$ is the hypothesis space of ReLU-activated neural network with L layers and the number of neurons at each layer is in the order of p. We give the following definition.

Definition 1. For the noise evaluation hypothesis h_θ, the risk of the hypothesis on a distribution $\mathcal{D}$ is defined by

the probability according to $\mathcal{D}$ that a processed hypothesis $I(g(h_{\boldsymbol{\theta}}(\boldsymbol{x})) > \tau)$ disagrees with a labeling function $\rho : \mathbb{R}^d \to \{0, 1\}$. Mathematically,

$$\varepsilon_{\mathcal{D}}(h) := \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\left[|I\left(g\left(h_{\boldsymbol{\theta}}(\boldsymbol{x})\right) > \tau\right) - \rho(\boldsymbol{x})|\right],$$

where $I(\cdot)$ is the indicator function, $\tau > 0$ is some threshold, and $\rho(\cdot)$ is the true labeling function.

Definition 1 is a form of the disagreement metric (Hanneke et al. 2014). Based on Definition 1 and results of (Ben-David et al. 2010; Bartlett et al. 2019), we can provide the following theoretical guarantee (proved in Appendix C) for the hard anomalous data $\tilde{\mathcal{X}}_H$.

Theorem 1. (Generalization Error Bound) Let $\varepsilon_{\tilde{\mathcal{D}}_H}(h)$ and $\varepsilon_{\hat{\mathcal{D}}}(h)$ be the risks of hypothesis h on $\tilde{\mathcal{D}}_H$ and $\hat{\mathcal{D}}$, respectively. If $\hat{\mathcal{X}}$ and $\tilde{\mathcal{X}}_H$ are unlabeled samples of size N each, drawn from $\hat{\mathcal{D}}$ and $\tilde{\mathcal{D}}_H$ respectively, then for any $\delta \in (0, 1)$, with probability at least $1 - \delta$, for every $h \in \mathcal{H}$:

$$\varepsilon_{\tilde{\mathcal{D}}_H}(h) \leq \varepsilon_{\hat{\mathcal{D}}}(h) + \tfrac{1}{2}\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\tilde{\mathcal{X}}_H, \hat{\mathcal{X}}) + \nu + \lambda,$$

where $\nu = 4\sqrt{\frac{2d_{vc}\log(2N)+\log(\frac{2}{\delta})}{N}}$ and $d_{vc} = \mathcal{O}(pL\log(pL))$. $\lambda = \varepsilon_{\hat{\mathcal{D}}}(h^*) + \varepsilon_{\tilde{\mathcal{D}}_H}(h^*)$ where h^* is the ideal joint hypothesis that minimize $\varepsilon_{\hat{\mathcal{D}}}(h) + \varepsilon_{\tilde{\mathcal{D}}_H}(h)$.

The definition of the divergence $\hat{d}_{\mathcal{H}\Delta\mathcal{H}}$ is shown in Appendix C for simplicity. As shown in Figure 1, Theorem 1 describes that if the divergence $\hat{d}_{\mathcal{H}\Delta\mathcal{H}}$ between the perturbed training data $\hat{\mathcal{X}}$ and the hard test data $\tilde{\mathcal{X}}_H$ is small, the model can correctly identify the anomaly. In other words, the generated data $\hat{\mathcal{X}}$ can be very useful for learning a reasonable detection model $h_{\boldsymbol{\theta}}$. Since there is no available information about $\tilde{\mathcal{X}}_H$ during the training, we cannot obtain the divergence $\hat{d}_{\mathcal{H}\Delta\mathcal{H}}$ in real practice. Hence, we always standardize the normal training data and make the added noise level relatively small. In the ablation study, we show that a large noise level is harmful to the training. Note that this guarantee does not apply to $\tilde{\mathcal{X}}_E$ because $\tilde{\mathcal{D}}_E$ may be very far from $\hat{\mathcal{D}}$. The following context will provide a guarantee for detecting $\tilde{\mathcal{X}}_E$. We make the following assumption and present the theoretical result (proved in Appendix C).

Assumption 1. For any $\boldsymbol{x}$ and $\tilde{\boldsymbol{x}}$ drawn from $\mathcal{D}$ and $\tilde{\mathcal{D}}_E$ respectively, there exists a constant $c > 0$ such that $c\|\boldsymbol{x} - \tilde{\boldsymbol{x}}\| \leq |g(h_{\boldsymbol{\theta}}(\boldsymbol{x})) - g(h_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}}))|$.

Theorem 2. Let $d_{\min} = \inf_{\boldsymbol{x}\sim\mathcal{D},\tilde{\boldsymbol{x}}\sim\tilde{\mathcal{D}}_E}\|\boldsymbol{x} - \tilde{\boldsymbol{x}}\|$ (shown by Figure 1). Suppose $g(h_{\boldsymbol{\theta}}(\boldsymbol{x})) < \epsilon$ for any $\boldsymbol{x} \in \mathcal{D}$, where $\epsilon \geq 0$. Then, under Assumption 1, any anomalous samples drawn from $\tilde{\mathcal{D}}_E$ can be successfully detected if $d_{\min} > \max\{\epsilon/c, \tau/c\}$.

This theorem provides a theoretical guarantee for our method to detect any anomalous sample in $\tilde{\mathcal{X}}_E$ or drawn from $\tilde{\mathcal{D}}_E$ (depicted by the blue double-headed arrow in Figure 1) as anomaly successfully. When c is larger, the detection is easier. We can derive a bound for c with more assumptions: if the learned model h is bijective and L-Lipschitz, then h^{-1} is $1/L$-Lipschitz, meaning that $c = \alpha L$, where $\alpha = \inf_{\boldsymbol{z}\in dom(g)}\|\nabla g(z)\|$. Using an invertible neural network (Behrmann et al. 2019) could achieve this. For an h with q layers, spectral norm $\|\mathbf{W}_i\|_\sigma$ for layer i, and ρ-Lipschitz activation, we have $c = \alpha\prod_{i=1}^{q}\rho^q\|\mathbf{W}_i\|_\sigma^q$. In our experiments, we used both MLP and ResMLP without dimensionality reduction. According to Theorem 1 of (Behrmann et al. 2019), ResMLP is invertible if each layer's Lipschitz constant is under 1, which is easy to ensure.

To sum up, Theorem 1 and Theorem 2 provide guarantees for detecting $\mathcal{X}_H$ and $\mathcal{X}_E$ respectively. Therefore, our method is theoretically guaranteed to detect $\tilde{\mathcal{X}}$, although we never use any real anomalous data in the training stage.

Experiments

Experimental Settings

Datasets We evaluate our method in two common settings: unsupervised anomaly detection and one-class classification. In the anomaly detection setting, where anomalous samples are few, we use 47 real-world tabular datasets[2] from (Han et al. 2022), covering domains like healthcare, image processing, and finance. For one-class classification setting, we collected 25 benchmark tabular datasets used in previous works (Pang et al. 2021; Shenkar and Wolf 2022). The raw data was sourced from the UCI Machine Learning Repository (Kelly, Longjohn, and Nottingham 0) and their official websites. For categoric value, we use a one-hot encoding. We test on all classes in multi-class datasets, reporting the average performance score per class, which is similar to one-class classification on image dataset (Cai and Fan 2022). For datasets with validation/testing sets, we train on all normal samples. If only a training set is available, we randomly split 50% of normal samples for training and use the rest with anomalous data for testing. Data is standardized using the training set's mean and standard deviation. Dataset details are in Table 4 and Table 5 of Appendix D.

Baseline Methods We select 12 baseline methods for comparative analysis, including probabilistic-based, proximity-based, deep neural network-based, ensemble-based methods, and recent UAD methods that can be applied to tabular data. They are Isolated Forest (**IForest**) (Liu, Ting, and Zhou 2008), **COPOD** (Li et al. 2020), auto-encoder (**AE**) (Aggarwal 2016), **KNN** (Angiulli and Pizzuti 2002), Local Outlier Factor (**LOF**) (Breunig et al. 2000), **DeepSVDD** (Ruff et al. 2018), **AnoGAN** (Schlegl et al. 2017), **ECOD** (Li et al. 2023), **SCAD** (Shenkar and Wolf 2022), **NeuTraLAD** (Qiu et al. 2021), **PLAD** (Cai and Fan 2022), and **DPAD** (Fu, Zhang, and Fan 2024). For DPAD, PLAD, SCAD, and NeuTraLAD, we use the code provided by the authors. For the other baseline methods, we utilize PyOD, a Python library developed by (Zhao, Nasrullah, and Li 2019). The default settings are adopted. We repeat 10 times for each baseline.

Implementation All experiments are implemented by Pytorch (Paszke et al. 2017) on NVIDIA Tesla V100 and Intel Xeon Gold 6200 platform. We utilize two network architectures for the evaluation, VanillaMLP (**MLP**) and **ResMLP**.

[2]https://github.com/Minqi824/ADBench/

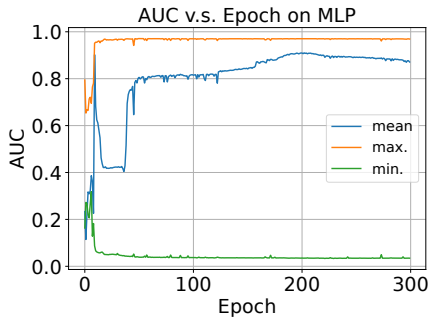


Figure 3: Comparison of different $g(\cdot)$, i.e. mean, maximum, and minimum, on KDD-CUP99, at each optimization epoch.

VanillaMLP is a plain ReLU-activated feed-forward neural network with 4 fully connected layers. ResMLP has a similar architecture to (Touvron et al. 2022) but has a simpler structure. Detailed architectures are in Appendix E. The network model is optimized by AMSGrad (Reddi, Kale, and Kumar 2018) with 10^{-4} learning rate and 5×10^{-4} weight decay. In the results, we adopt Gaussian noise in the noise generation, maximum noise level $\sigma_{max} = 2$, and the number of different noise distributions $m = 3$. To enlarge the noised sample number, we generate 3 noise-augmented instances for the same input instance with 3 different noise ratios, 0.5, 0.8, and 1.0 at each epoch. Unless specified, we train the model for 500 epochs and manually decay the learning rate at the 100-th epoch in a factor of 0.1.

Performance Metric Since the output of h is a noise evaluation vector, an operation $g(\cdot)$ is introduced, as explained by (10). Here we select the maximum. A comparative result on KDD-CUP99 among maximum, minimum, and mean is shown in Figure 3 where we train the model for 300 epochs. The maximum reaches the highest performance and the fastest convergence speed. For the performance metric, we report the area under the ROC curve (AUC) and F1 score. The calculation of the F1 score and the determination of anomaly threshold τ are consistent with (Shenkar and Wolf 2022; Qiu et al. 2021).

Unsupervised Anomaly Detection Results

The average result under the unsupervised anomaly detection setting of ten runs is reported in Figure 4. We conducted a comparative analysis of 11 tabular UAD methods across 47 datasets, evaluating average AUC, average F1, average ranking in AUC, and average ranking in F1. It is seen that our methods not only significantly outperform the AE methods, but also reach the highest ranking. The detailed results on each dataset and p-value from paired t-test are reported in Appendix F (Tables 6 and 7), which emphasizes the statistical significance of the improvements achieved by our methods.

One-Class Classification Results

We compared our noise evaluation method with 12 baseline methods on the OCC dataset setting. The results in Table 3 show that our anomaly detection techniques, Ours-ResMLP and Ours-MLP, consistently outperform baselines across various tabular datasets, with mean AUC values of 92.68 ± 11.10 and 92.27 ± 11.1, indicating greater effectiveness and lower variability. While traditional methods like IForest and KNN perform well, our methods excel, especially on complex datasets like musk and optdigits. Though our methods may not always lead in AUC, the difference is minimal, around 1%. For faster inference, the MLP model is recommended. The last two lines in the table are the p-values of paired t-test of our two methods (ResMLP and MLP) against each baseline method. The paired t-test is based on 25 pairs (as there are 25 datasets). Our approach also achieved the highest average F1 score and rank, 94.18% and 1.52, as detailed in Appendix G.

Type	Parameter	Value	Offset
Gaussian	μ, σ	$\mu = 0, \sigma = \hat{\sigma}$	0
Laplace	μ, σ	$\mu = 0, \sigma = \hat{\sigma}$	0
Uniform	a, b	$a = -\sqrt{3}\hat{\sigma}, b = \sqrt{3}\hat{\sigma}$	0
Rayleigh	s	$s = \sqrt{\frac{2}{4-\pi}}\hat{\sigma}$	$-\sqrt{\frac{\pi}{4-\pi}}\hat{\sigma}$
Gamma	α, β	$\alpha = \sqrt{\frac{1}{\beta}}\hat{\sigma}, \beta = \beta$	$-\sqrt{\beta}\hat{\sigma}$
Poisson	λ	$\lambda = \hat{\sigma}$	$-\hat{\sigma}$

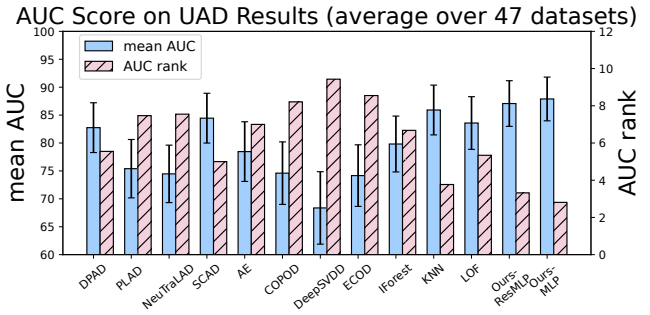
Table 2: Eight different noise types are adopted. We adjust the parameter to make it 0 mean value and $\hat{\sigma}$ standard deviation, where $\hat{\sigma}$ is the designated noise level. If the noise is non-negative by default, we offset its mean to 0. For the Gamma noise, β is a non-negative hyper-parameter, where we evaluate $\beta = 1$ and $\beta = 3$. For Salt&Pepper and Bernoulli noise, we generate a probability vector based on a uniform distribution. Then, generate a binary vector using the probability vector. If there is noise, we change the value into the maximum or minimum value in the batch or flip the sign of the element of $\boldsymbol{x}$.

Compared with many popular UAD methods, our noise evaluation method has the following advantages:

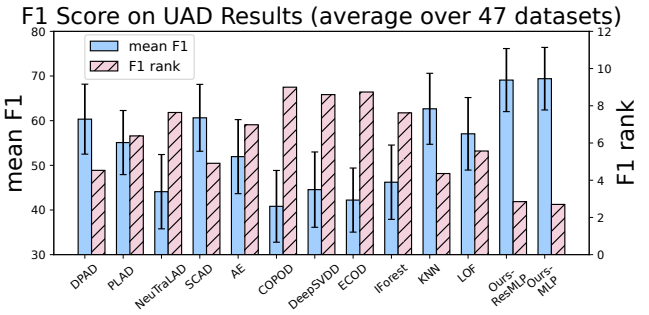
- Noise evaluation shows superior performance in the extensive empirical experiments, indicating our method can accurately identify anomalies in tabular data.
- The generalization and anomaly detection ability of noise evaluation can be theoretically guaranteed, whereas many deep learning based methods lack such guarantees (Hussain et al. 2023).
- The implementation of our method is easier. The perturbed sample generation can be pre-generated without extra training. In addition, no hyper-parameter is tuned in the learning objective.

Ablation Study

As discussed in the Proposed Method Section, we have several alternatives. In this section, we study how different noise levels, noise ratios, and noise types affect the performance of our method. We utilize the OCC setting to perform the ablation study. Detailed results are shown in Appendix H.



(a) AUC and Rank on UAD Results.

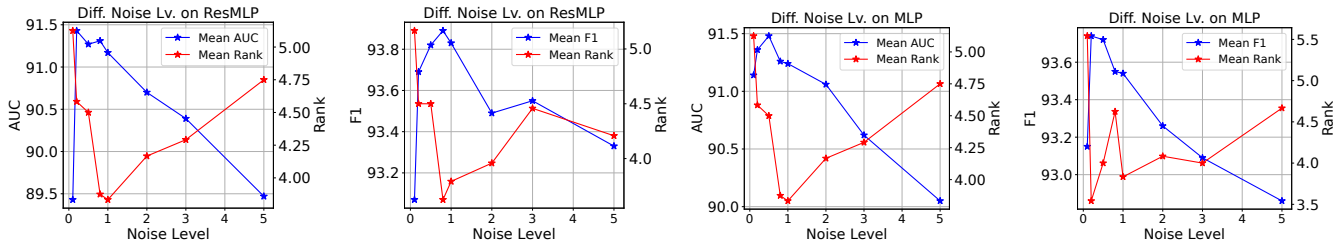


(b) F1 and Rank on UAD Results.

Figure 4: AUC (%) and F1 (%) score of the proposed method compared with 11 baselines on 47 benchmark datasets. Each experiment is repeated 10 times with random seed from 0 to 9, and mean value and 95% confidence interval are reported. Rank (the lower the better) is calculated out of 12 tested methods.

Methods	DPAD (2024)	PLAD (2022)	NeuTraLAD (2021)	SCAD (2022)	AE	AnoGAN	COPOD	DeepSVDD	ECOD (2023)	IForest	KNN	LOF	Ours-ResMLP	Ours-MLP
abalone	70.64±13.2	64.23±15.4	45.29±8.8	63.90±14.7	71.69±11.1	68.65±14.7	43.57±15.0	59.78±16.1	45.00±16.9	71.55±12.2	69.11±14.1	68.89±13.0	**76.90**±11.0	**76.78**±10.9
arrhyth.	76.38±1.0	63.15±5.1	52.13±0.0	69.60±2.0	76.12±1.8	69.61±2.6	75.23±1.4	72.82±2.2	75.21±1.5	77.06±1.8	75.87±2.0	75.85±1.9	**77.34**±2.4	76.77±2.1
breastw	98.36±0.3	87.4±15.0	78.77±3.0	97.29±0.7	98.83±0.4	99.11±0.3	**99.29**±0.2	95.58±1.1	98.62±0.3	99.36±0.2	98.84±0.3	95.43±1.6	98.49±0.7	98.84±0.4
cardio	82.13±3.3	74.82±8.9	55.84±2.6	69.37±1.8	83.36±0.8	76.78±12.2	65.98±0.7	53.12±7.7	78.03±0.6	81.74±2.0	73.96±1.0	77.37±1.5	**85.59**±2.4	**86.01**±3.3
ecoli	92.25±4.4	68.40±2.6	48.46±12.0	88.17±5.7	93.18±4.0	90.86±6.2	49.16±22.2	87.63±5.8	50.98±11.6	93.21±3.6	93.19±4.2	91.42±5.7	92.98±3.7	**93.53**±3.7
glass	76.73±7.4	64.8±10.5	57.68±9.0	78.89±6.0	73.39±10.4	74.76±11.0	44.61±26.1	75.39±8.2	44.75±24.2	76.96±10.5	76.66±9.0	71.09±9.7	**83.12**±10.1	**84.41**±10.2
ionosph.	96.66±0.4	64.26±15.2	90.77±2.4	96.83±0.7	90.40±1.2	86.19±5.9	80.05±1.6	95.56±1.1	74.33±1.6	89.53±2.9	97.47±0.8	95.44±1.3	**97.53**±0.3	97.11±0.7
kdd	81.76±11.9	90.87±6.6	93.6±1.7	91.56±4.0	95.12±3.3	92.71±2.5	76.51±1.2	72.81±27.8	78.75±1.3	96.12±0.3	94.45±0.2	85.05±0.2	**95.69**±1.2	**96.92**±0.6
letter	93.39±3.3	59.05±15.3	90.95±4.6	99.26±0.5	88.91±4.3	80.23±8.5	50.03±15.8	94.25±2.4	50.09±15.4	91.15±3.4	97.95±1.6	95.61±2.6	**99.42**±0.3	**99.42**±0.4
lympho	74.21±3.6	64.96±8.1	46.6±2.2	74.91±4.6	75.84±3.7	70.15±6.7	53.38±3.7	73.37±5.8	53.52±3.5	75.14±5.3	77.86±3.2	76.50±3.2	**81.14**±4.7	**82.37**±5.0
mammo.	88.32±1.7	82.38±2.3	65.57±2.3	78.18±2.2	85.65±1.1	78.67±13.4	90.54±0.1	69.87±7.1	**90.63**±0.1	87.91±0.7	87.55±0.3	84.12±1.2	90.11±0.9	89.89±0.4
mulcross	**100.0**±0.0	99.90±0.0	76.59±11.3	**100.0**±0.0	**100.0**±0.0	99.89±0.1	93.24±0.0	**100.0**±0.0	95.97±0.1	99.90±0.1	**100.0**±0.0	**100.0**±0.0	**100.0**±0.0	**100.0**±0.0
musk	71.36±2.9	74.42±3.3	78.86±1.6	81.30±0.7	30.68±0.6	41.37±12.2	32.03±0.4	62.02±6.0	30.16±0.4	46.74±3.7	81.94±0.4	81.31±0.6	**85.06**±0.5	**83.69**±1.0
optdigits	93.43±4.3	80.43±15.5	67.3±15.0	97.80±1.8	95.69±2.6	90.04±6.6	71.72±8.8	94.86±3.7	68.69±10.1	97.55±1.7	97.25±2.2	97.17±2.3	**97.93**±1.7	**98.00**±1.7
pendigits	98.04±2.1	86.45±14.9	97.23±2.4	99.65±0.3	95.46±4.7	92.56±9.8	49.90±25.2	94.25±5.3	49.93±22.9	98.08±1.7	99.59±0.5	99.11±1.0	**99.82**±0.2	**99.84**±0.2
pima	70.5±1.7	63.67±5.7	48.63±4.6	67.97±1.3	69.73±1.5	70.94±5.0	65.58±1.0	58.18±2.9	59.46±1.3	73.33±1.4	**74.63**±1.0	71.06±1.5	72.42±2.6	73.53±2.9
satimage	92.83±5.1	54.83±19.4	47.98±12.0	93.35±4.8	93.39±5.4	91.98±6.8	69.24±26.8	75.10±7.7	62.75±22.8	**95.33**±3.8	94.85±4.4	89.67±6.6	94.51±3.6	93.72±4.7
seismic	**74.81**±0.8	72.59±2.8	67.75±1.2	72.47±0.8	71.13±0.5	72.82±1.5	73.87±0.5	54.32±10.2	70.17±0.4	73.48±0.6	74.57±0.5	60.87±1.6	71.38±1.3	72.62±1.9
shuttle	98.93±1.4	88.55±14.2	97.42±2.4	98.30±2.0	92.49±8.8	88.42±16.2	31.08±31.4	96.51±6.5	35.53±30.8	91.55±6.7	98.66±1.4	95.60±6.2	**99.35**±0.8	**99.40**±1.0
speech	54.38±2.7	57.32±4.9	56.31±4.8	57.85±2.4	47.08±0.5	50.33±5.4	49.15±0.6	52.20±3.8	47.08±0.5	46.88±1.6	48.67±0.7	49.81±0.5	57.14±2.6	**58.38**±2.1
thyroid	91.96±1.0	97.02±1.7	80.02±3.4	86.53±1.4	83.38±0.6	72.63±4.3	78.46±0.0	57.35±3.5	79.17±0.0	90.08±1.4	91.71±0.0	88.02±0.0	**97.47**±0.1	**96.91**±0.2
vertebral	87.31±2.6	56.47±28.6	50.35±3.5	78.56±3.3	88.58±1.3	87.96±3.2	78.63±2.0	80.43±3.8	60.02±2.9	86.08±2.3	87.18±1.0	88.20±1.5	**90.91**±1.4	**89.91**±1.2
vowels	80.22±4.1	79.77±17.1	97.25±1.1	**99.52**±0.2	62.80±0.8	59.32±7.8	49.73±0.8	72.01±6.5	59.39±0.7	78.17±2.2	97.16±0.4	95.53±0.7	99.42±0.3	99.14±0.2
wbc	95.10±1.1	68.07±15.0	53.01±12.6	93.11±1.3	95.63±0.6	94.01±2.5	86.53±1.0	90.22±3.3	62.46±1.4	95.74±0.8	94.68±0.9	95.09±0.9	**96.98**±1.1	96.73±1.1
wine	95.68±4.5	67.13±7.8	43.95±12.0	86.21±7.8	97.00±4.1	94.47±10.4	49.37±7.8	95.46±5.5	49.37±11.5	96.23±3.1	97.81±2.8	97.98±2.4	**98.76**±1.8	**98.33**±2.3
mean	88.05±12.4	67.42±19.6	71.45±22.6	88.59±15.3	85.68±13.2	80.17±14.8	53.38±22.4	80.85±19.1	52.77±20.9	87.0±12.8	90.37±13.5	88.72±13.4	**92.68**±11.0	**92.27**±11.1
m. rank	4.96	9.16	10.16	5.96	6.00	8.12	10.12	8.68	10.40	4.96	3.92	5.88	**2.04**	**1.68**
p-value (Res.)	0.0003	0.0000	0.00000	0.0001	0.0066	0.0002	0.0000	0.0000	0.0000	0.0054	0.0025	0.0000	-	0.47
p-value (MLP)	0.0002	0.0000	0.0000	0.0001	0.0051	0.0002	0.0000	0.0000	0.0000	0.0036	0.0022	0.0000	0.47	-

Table 3: AUC (%) score of the proposed method compared with 12 baselines on 25 benchmark datasets. Each experiment is repeated 10 times with random seed from 0 to 9, and mean value and standard deviation are reported. Mean is the average AUC score under all experiments. Mean rank (the lower the better) is calculated out of 10 tested methods. Mammo. refers to mammography.



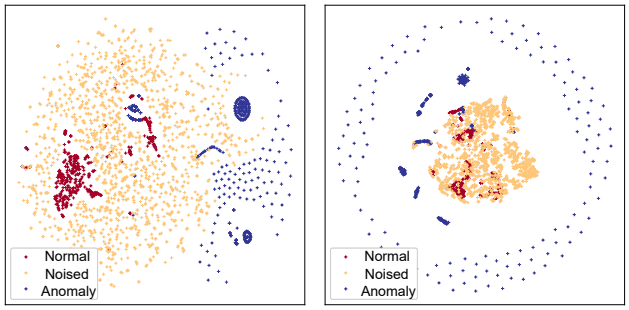
(a) AUC and Its Rank on ResMLP with Different Noise Levels (b) F1 and Its Rank on ResMLP with Different Noise Levels (c) AUC and Its Rank on MLP with Different Noise Levels (d) F1 and Its Rank on MLP with Different Noise Levels

Figure 5: Sensitivity of Different Noise Level in $[0.1, 0.2, 0.5, 0.8, 1.0, 2.0, 3.0, 5.0]$. The mean rank (the lower the better) is calculated out of 8 noise levels.

Sensitivity of Different Noise Levels To explore how different noise levels affect performance, we use Gaussian noise and generate 3 noised instances per training batch, consistent with previous settings. We test noise levels in $[0.1, 0.2, 0.5, 0.8, 1.0, 2.0, 3.0, 5.0]$. The results are reported in Figure 5. Results show that too small noise levels confuse the model due to the minimal distance between normal samples and anomalies, while too high noise levels expand the output value range and sampling space, reducing effectiveness as theorem 1 suggested. Hence, the optimal noise level is around 1.0.

Sensitivity of Different Noise Types We explore different noise types with a mean of 0 and a standard deviation (noise level) of σ. We use Salt&Pepper noise, Gaussian, Laplace, Uniform, Rayleigh, Gamma, Poisson, and Bernoulli distributions. For Salt&Pepper and Bernoulli noise, a probability vector from a uniform distribution generates a binary vector, dictating feature value alterations by changing values or reversing signs. Other distributions are adjusted to have zero mean and σ standard deviation. Detailed parameters are in described in Table 2, with results in Figure 10 of Appendix H. Results show Salt&Pepper and Bernoulli noise performs poorly, indicating the effectiveness of our noise generation design. Gaussian, Rayleigh, and Uniform noise maintain stable performance with 92% AUC and 94% F1 scores, indicating our method's robustness with various noise types.

Qualitative Visualization We utilize t-Distributed Stochastic Neighbor Embedding (t-SNE) visualization (Van der Maaten and Hinton 2008) to show the allocation of normal, noised, and true anomalous samples in the neural network embedding space qualitatively. In Figure 6, we visualize the penultimate layer of the neural network using KDD-CUP99. It is seen that the introduction of noised instances into the dataset ostensibly aids the model in constructing a discriminative boundary that proficiently segregates the in-distribution data from its out-of-distribution counterparts. Empirical observations reveal that the actual anomalous data predominantly falls outside of this established boundary, a phenomenon consistently manifested in the experimental results for both ResMLP and MLP models.



(a) ResMLP T-SNE Visual. (b) MLP T-SNE Visualization

Figure 6: Qualitative visualization of the penultimate layer on the KDD-CUP99 dataset shows normal (red), noised (yellow), and anomalous (blue) instances. The noised instances help the model establish a decision boundary between in-distribution and out-of-distribution data. Real anomalies consistently fall outside this boundary in both ResMLP and MLP, confirming their effectiveness in anomaly detection.

Conclusion

In conclusion, we presented a novel noise evaluation-based method for unsupervised anomaly detection in tabular data. We assume the model can learn the anomalous pattern from the noised normal data. Predicting the magnitude of the noise shows inspiration for how much and where the abnormality is. We theoretically proved the generalizability and reliability of our method. Extensive experiments demonstrated that our approach outperforms other anomaly detection methods through 47 real tabular datasets in the UAD setting and 25 real tabular datasets in the OCC setting. An ablation study suggests that using Gaussian, Rayleigh, and Uniform noise has a stable performance. One potential limitation of this work is that we focused on tabular data only. Nevertheless, it is possible to extend our method to image data via the following two steps: 1) extract the features of input images using a pretrained encoder; 2) apply our method to the extracted image features. This could be interesting future work. The full version of the paper with appendices can be found at https://arxiv.org/abs/2412.11461.

Acknowledgments

This work was supported by the Shenzhen Science and Technology Program under the Grant No.JCYJ20210324130208022 (Fundamental Algorithms of Natural Language Understanding for Chinese Medical Text Processing) and the General Program of Guangdong Basic and Applied Basic Research under Grant No.2024A1515011771.

References

Abououf, M.; Singh, S.; Mizouni, R.; and Otrok, H. 2023. Explainable AI for Event and Anomaly Detection and Classification in Healthcare Monitoring Systems. *IEEE Internet of Things Journal*.

Aggarwal, C. C. 2016. Outlier analysis second edition.

Angiulli, F.; and Pizzuti, C. 2002. Fast outlier detection in high dimensional spaces. In *European conference on principles of data mining and knowledge discovery*, 15–27. Springer.

Bartlett, P. L.; Harvey, N.; Liaw, C.; and Mehrabian, A. 2019. Nearly-tight VC-dimension and pseudodimension bounds for piecewise linear neural networks. *The Journal of Machine Learning Research*, 20(1): 2285–2301.

Behrmann, J.; Grathwohl, W.; Chen, R. T.; Duvenaud, D.; and Jacobsen, J.-H. 2019. Invertible residual networks. In *International conference on machine learning*, 573–582. PMLR.

Ben-David, S.; Blitzer, J.; Crammer, K.; Kulesza, A.; Pereira, F.; and Vaughan, J. W. 2010. A theory of learning from different domains. *Machine learning*, 79: 151–175.

Breunig, M. M.; Kriegel, H.-P.; Ng, R. T.; and Sander, J. 2000. LOF: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, 93–104.

Cai, J.; and Fan, J. 2022. Perturbation learning based anomaly detection. *Advances in Neural Information Processing Systems*, 35.

Chang, C.-H.; Yoon, J.; Arik, S. Ö.; Udell, M.; and Pfister, T. 2023. Data-efficient and interpretable tabular anomaly detection. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 190–201.

Chen, J.; Sathe, S.; Aggarwal, C.; and Turaga, D. 2017. Outlier detection with autoencoder ensembles. In *Proceedings of the 2017 SIAM international conference on data mining*, 90–98. SIAM.

Chen, Y.; Tian, Y.; Pang, G.; and Carneiro, G. 2022. Deep one-class classification via interpolated gaussian descriptor. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 383–392.

Ding, X.; Zhao, L.; and Akoglu, L. 2022. Hyperparameter sensitivity in deep outlier detection: Analysis and a scalable hyper-ensemble solution. *Advances in Neural Information Processing Systems*, 35: 9603–9616.

Fan, J.; Chow, T. W. S.; and Qin, S. J. 2022. Kernel-Based Statistical Process Monitoring and Fault Detection in the Presence of Missing Data. *IEEE Transactions on Industrial Informatics*, 18(7): 4477–4487.

Fu, D.; Zhang, Z.; and Fan, J. 2024. Dense projection for anomaly detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 8398–8408.

Goyal, S.; Raghunathan, A.; Jain, M.; Simhadri, H. V.; and Jain, P. 2020. DROCC: Deep robust one-class classification. In *International conference on machine learning*, 3711–3721. PMLR.

Han, S.; Hu, X.; Huang, H.; Jiang, M.; and Zhao, Y. 2022. Adbench: Anomaly detection benchmark. *Advances in Neural Information Processing Systems*, 35: 32142–32159.

Hanneke, S.; et al. 2014. Theory of disagreement-based active learning. *Foundations and Trends® in Machine Learning*, 7(2-3): 131–309.

He, Z.; Xu, X.; and Deng, S. 2003. Discovering cluster-based local outliers. *Pattern recognition letters*, 24(9-10): 1641–1650.

Hilal, W.; Gadsden, S. A.; and Yawney, J. 2022. Financial fraud: a review of anomaly detection techniques and recent advances. *Expert systems With applications*, 193: 116429.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33: 6840–6851.

Hu, T.; Zhang, J.; Yi, R.; Du, Y.; Chen, X.; Liu, L.; Wang, Y.; and Wang, C. 2024. Anomalydiffusion: Few-shot anomaly image generation with diffusion model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 8526–8534.

Huang, Y.; Zhang, Y.; Wang, L.; Zhang, F.; and Lin, X. 2024. EntropyStop: Unsupervised Deep Outlier Detection with Loss Entropy. *ACM KDD*.

Hussain, M.; Suh, J.-W.; Seo, B.-S.; and Hong, J.-E. 2023. How Reliable are the Deep Learning-based Anomaly Detectors? A Comprehensive Reliability Analysis of Autoencoder-based Anomaly Detectors. In *2023 Fourteenth International Conference on Ubiquitous and Future Networks (ICUFN)*, 317–322. IEEE.

Jin, M.; Liu, Y.; Zheng, Y.; Chi, L.; Li, Y.-F.; and Pan, S. 2021. Anemone: Graph anomaly detection with multi-scale contrastive learning. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 3122–3126.

Kelly, M.; Longjohn, R.; and Nottingham, K. 0. The UCI Machine Learning Repository. Accessed: 2023-12-27.

Li, Z.; Zhao, Y.; Botta, N.; Ionescu, C.; and Hu, X. 2020. COPOD: copula-based outlier detection. In *2020 IEEE international conference on data mining (ICDM)*, 1118–1123. IEEE.

Li, Z.; Zhao, Y.; Hu, X.; Botta, N.; Ionescu, C.; and Chen, G. H. 2023. ECOD: Unsupervised Outlier Detection Using Empirical Cumulative Distribution Functions. *IEEE Transactions on Knowledge and Data Engineering*, 35(12): 12181–12193.

Liu, F. T.; Ting, K. M.; and Zhou, Z.-H. 2008. Isolation forest. In *2008 eighth ieee international conference on data mining*, 413–422. IEEE.

Liu, Y.; Li, Z.; Zhou, C.; Jiang, Y.; Sun, J.; Wang, M.; and He, X. 2019. Generative adversarial active learning for unsupervised outlier detection. *IEEE Transactions on Knowledge and Data Engineering*, 32(8): 1517–1528.

Pang, G.; Shen, C.; Cao, L.; and Hengel, A. V. D. 2021. Deep learning for anomaly detection: A review. *ACM computing surveys (CSUR)*, 54(2): 1–38.

Papadimitriou, S.; Kitagawa, H.; Gibbons, P. B.; and Faloutsos, C. 2003. Loci: Fast outlier detection using the local correlation integral. In *Proceedings 19th international conference on data engineering (Cat. No. 03CH37405)*, 315–326. IEEE.

Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; and Lerer, A. 2017. Automatic differentiation in pytorch. *NIPS 2017 Workshop*.

Qiu, C.; Pfrommer, T.; Kloft, M.; Mandt, S.; and Rudolph, M. 2021. Neural transformation learning for deep anomaly detection beyond images. In *International conference on machine learning*, 8703–8714. PMLR.

Reddi, S. J.; Kale, S.; and Kumar, S. 2018. On the Convergence of Adam and Beyond. In *International Conference on Learning Representations*.

Roth, K.; Pemula, L.; Zepeda, J.; Schölkopf, B.; Brox, T.; and Gehler, P. 2022. Towards total recall in industrial anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14318–14328.

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

Saeed, M. M.; Saeed, R. A.; Abdelhaq, M.; Alsaqour, R.; Hasan, M. K.; and Mokhtar, R. A. 2023. Anomaly Detection in 6G Networks Using Machine Learning Methods. *Electronics*, 12(15): 3300.

Schlegl, T.; Seeböck, P.; Waldstein, S. M.; Schmidt-Erfurth, U.; and Langs, G. 2017. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In *International conference on information processing in medical imaging*, 146–157. Springer.

Schölkopf, B.; Platt, J. C.; Shawe-Taylor, J.; Smola, A. J.; and Williamson, R. C. 2001. Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7): 1443–1471.

Shenkar, T.; and Wolf, L. 2022. Anomaly detection for tabular data with internal contrastive learning. In *International Conference on Learning Representations*.

Siddiqui, M. A.; Stokes, J. W.; Seifert, C.; Argyle, E.; McCann, R.; Neil, J.; and Carroll, J. 2019. Detecting cyber attacks using anomaly detection with explanations and expert feedback. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2872–2876. IEEE.

Sohn, K.; Li, C.-L.; Yoon, J.; Jin, M.; and Pfister, T. 2020. Learning and Evaluating Representations for Deep One-Class Classification. In *International Conference on Learning Representations*.

Song, Y.; and Ermon, S. 2019. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32.

Tax, D. M.; and Duin, R. P. 2004. Support vector data description. *Machine learning*, 54: 45–66.

Touvron, H.; Bojanowski, P.; Caron, M.; Cord, M.; El-Nouby, A.; Grave, E.; Izacard, G.; Joulin, A.; Synnaeve, G.; Verbeek, J.; et al. 2022. Resmlp: Feedforward networks for image classification with data-efficient training. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4): 5314–5321.

Tur, A. O.; Dall'Asen, N.; Beyan, C.; and Ricci, E. 2023. Exploring diffusion models for unsupervised video anomaly detection. In *2023 IEEE International Conference on Image Processing (ICIP)*, 2540–2544. IEEE.

Van der Maaten, L.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11).

Vincent, P. 2011. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7): 1661–1674.

Wang, S.; Wang, X.; Zhang, L.; and Zhong, Y. 2021. Auto-AD: Autonomous hyperspectral anomaly detection network based on fully convolutional autoencoder. *IEEE Transactions on Geoscience and Remote Sensing*, 60: 1–14.

Xiao, F.; Zhou, J.; Han, K.; Hu, H.; and Fan, J. 2025. Unsupervised anomaly detection using inverse generative adversarial networks. *Information Sciences*, 689: 121435.

Yan, X.; Zhang, H.; Xu, X.; Hu, X.; and Heng, P.-A. 2021. Learning semantic context from normal samples for unsupervised anomaly detection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, 3110–3118.

Yang, X.; Latecki, L. J.; and Pokrajac, D. 2009. Outlier detection with globally optimal exemplar-based GMM. In *Proceedings of the 2009 SIAM international conference on data mining*, 145–154. SIAM.

Yang, X.; Qi, X.; and Zhou, X. 2023. Deep Learning Technologies for Time Series Abnormality Detection in Healthcare: A Review. *IEEE Access*.

Zavrtanik, V.; Kristan, M.; and Skočaj, D. 2021. Draem-a discriminatively trained reconstruction embedding for surface anomaly detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, 8330–8339.

Zhang, X.; Li, N.; Li, J.; Dai, T.; Jiang, Y.; and Xia, S.-T. 2023. Unsupervised surface anomaly detection with diffusion probabilistic model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 6782–6791.

Zhang, Y.; Sun, Y.; Cai, J.; and Fan, J. 2024. Deep Orthogonal Hypersphere Compression for Anomaly Detection. In *Proceedings of the International Conference on Learning Representations*.

Zhao, Y.; Nasrullah, Z.; and Li, Z. 2019. PyOD: A Python Toolbox for Scalable Outlier Detection. *Journal of Machine Learning Research*, 20(96): 1–7.

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