Dense Projection for Anomaly Detection

Dazhi Fu[1,2], **Zhao Zhang**[3], **Jicong Fan**[1,4*]

[1] The Chinese University of Hong Kong, Shenzhen, China
[2] University of Electronic Science and Technology of China, Chengdu, China
[3] Hefei University of Technology, Hefei, China
[4] Shenzhen Research Institute of Big Data, Shenzhen, China
fudazhiaka@gmail.com, cszzhang@gmail.com, fanjicong@cuhk.edu.cn

Abstract

This work presents a novel method called **d**ense **p**rojection for unsupervised **a**nomaly **d**etection (DPAD). The main idea is maximizing the local density of (normal) training data and then determining whether a test data is anomalous or not by evaluating its density. Specifically, DPAD uses a deep neural network to learn locally dense representations of normal data. Since density estimation is computationally expensive, we minimize the local distances of the representations in an iteratively reweighting manner, where the weights are updated adaptively and the parameters are regularized to avoid model collapse (all representations collapse to a single point). Compared with many state-of-the-art methods of anomaly detection, our DPAD does not rely on any assumption about the distribution or spatial structure of the normal data and representations. Moreover, we provide theoretical guarantees for the effectiveness of DPAD. The experiments show that our method DPAD is effective not only in traditional one-class classification problems but also in scenarios with complex normal data composed of multiple classes.

Introduction

Anomaly detection (Chandola, Banerjee, and Kumar 2009; Pang et al. 2021; Ruff et al. 2021; Cai and Fan 2022; Xiao, Sun, and Fan 2023) is an important problem in many areas such as machine learning, computer vision, medical imaging, and other fields (Fan and Wang 2014; Fan, Wang, and Zhang 2017). Basically, anomaly detection is a task that aims to identify anomalous data from normal data within a given dataset. To better simulate real-world scenarios, anomalous data is often considered to be unknown in the training stage, making this task typically an unsupervised learning problem. In the past decades, numerous anomaly detection methods have been proposed. In general, we can categorize them into three main types: density-based methods, reconstruction-based methods, and one-class classification methods, though there are other types such as the perturbation learning based method proposed by (Cai and Fan 2022).

Density-based methods assume that normal data occur in high-density regions, while anomalies are located in low-density or sparse regions, and utilize probabilistic models to model the distribution of normal data. Thus, popular density estimation methods such as kernel density estimation (KDE) Parzen (1962) and Gaussian mixture models (GMM) can be applied to anomaly detection. K-nearest-neighbors (kNN) is also a density-based method where the average distance from test data to its nearest k neighbors is measured as the anomaly score. This method relies heavily on the choice of k and may not be effective in handling high-dimensional data. kNN+ (Sun et al. 2022), utilizing a pre-trained neural network to learn feature embeddings of normal data, assumes that the test anomalies are relatively far away from the normal data and detects anomalies by using kNN in the embedding space, which makes it effective when faced with complex data. Breunig et al. (2000) proposed a method called local outlier factor (LOF), which relies on the concept that anomalous data often lie in a region of lower density than its surrounding data points. Zong et al. (2018) proposed deep autoencoding Gaussian mixture models (DAGMM) that combines deep auto-encoders with GMM, where the output energy generated by the GMM is used as an anomaly score. Deecke et al. (2019) provided an anomaly detection method ADGAN based on adversarial networks (GAN (Goodfellow et al. 2014)). ADGAN utilizes a generator to learn the distribution of normal data and a discriminator to detect anomalous data.

Reconstruction-based methods use neural networks such as auto-encoder (AE) to learn low-dimensional representation to reconstruct input data and utilize the reconstruction error as a metric to discern anomalies from normal instances. Auto-encoder and its various variants (Hinton and Salakhutdinov 2006; Vincent et al. 2008; Pidhorskyi, Almohsen, and Doretto 2018; Wang et al. 2021) consist of an encoder and a decoder, where the encoder compresses the input data into a latent effective representation, while the decoder reconstructs the original data from the compressed representation. These methods often rely on the assumption that normal data can be reconstructed effectively, while anomalous data exhibits significantly higher reconstruction errors. However, in practice, some anomalous samples can be well-reconstructed by auto-encoders, especially when the model is complex.

One-class classification methods train classifiers using only normal data. For instance, the one-class support vector machine (OC-SVM), proposed by (Schölkopf et al. 2001),

*Corresponding author

assumes that normal data can be separated from the rest of the data by a hyperplane in a high-dimensional feature space and tries to maximize the margin between the hyperplane and origin. Tax and Duin (2004) proposed support vector data description (SVDD) that aims to obtain a hypersphere with the smallest volume that encloses the normal data points while keeping the abnormal data points outside the hypersphere. To handle more complex data, Ruff et al. (2018) proposed Deep SVDD, which is based on an integration of deep learning (LeCun, Bengio, and Hinton 2015) and SVDD. Deep SVDD utilizes deep neural networks to learn effective feature embeddings from the normal data while aiming to enclose the normal data within a hypersphere with minimum volume. To ensure that any example reconstructed from the learned representation is normal data, Perera, Nallapati, and Xiang (2019) proposed one-class GAN (OCGAN), which trains an auto-encoder and discriminator adversarially. Goyal et al. (2020) presented a method called deep robust one-class classification (DROCC), which assumes that normal data resides in a low-dimensional manifold structure. It constructs anomalous samples in the training stage and classifies a point as anomalous if it is outside the union of balls around training data. This approach has been shown to be effective on various datasets. Hu et al. (2020) proposed H-Regularization with 2-Norm instance level normalization (HRN), including new loss function (called one-class loss), holistic regularization, and normalization, which can directly learn from a single class of data. Chen et al. (2022) proposed a method called interpolated Gaussian descriptor (IGD). It learns effective normality description based on representative normal data instead of fringe edge normal data.

It is worth noting that, density-based methods are not effective in handling high-dimensional data, reconstruction-based methods often suffer from overfitting, and one-class classification methods may not obtain their assumed reliable decision boundaries such as hypersphere. To address these limitations all at once, in this work, we propose a new density-based method called **D**ense **P**rojection for **A**nomaly **D**etection (DPAD). The main idea of DPAD is to train a neural network to learn a locally dense low-dimensional representation of normal data by reducing the distance between the representations of similar data (see Figure 1), and then density-based methods such as KNN can be applied to the representation to detect anomalies. Our contributions are summarized as follows:

- We propose a novel density-based method called DPAD for unsupervised anomaly detection. DPAD does not rely on any assumption about the shape of the decision boundary between normal data and anomalous data and is able to handle high-dimensional data effectively
- We propose to increase the local density of the region where normal data resides by reducing the distance between similar normal data locally.
- We thoroughly evaluate the effectiveness of dimensionality reduction plus KNN in unsupervised anomaly detection.
- In addition to experiments on classical one-class classification, we conduct challenging experiments where normal data are composed of multiple classes to further investigate the performance of DPAD and other methods.

Related Work

Before elaborating on our DPAD, we discuss the connection and difference between our DPAD and existing dimensionality reduction methods and DeepSVDD (Ruff et al. 2018).

Dimensionality Reduction + kNN

Dimensionality reduction (DR) methods are commonly used to address challenges such as the curse of dimensionality, data redundancy, and high computational complexity (Fan et al. 2018; Sun, Han, and Fan 2023). The best-known DR method is the principal component analysis (PCA) (Jolliffe and Cadima 2016). PCA is a linear DR method and is not effective in handling data with nonlinear structures. There have been many nonlinear DR methods, e.g., LLE (Roweis and Saul 2000), Isomap (Tenenbaum, Silva, and Langford 2000), AE (Hinton and Salakhutdinov 2006), t-SNE (Van der Maaten and Hinton 2008), and UMAP (McInnes, Healy, and Melville 2018). Particularly, AE is more useful in feature extraction while t-SNE and UMAP are more useful in 2D visualization. AE solves the following problem $\min_{f,g} \mathbb{E}_{\mathbf{x}\sim\mathcal{D}}[\|\mathbf{x} - g(f(\mathbf{x}))\|_\ell]$ where $f : \mathbb{R}^D \to \mathbb{R}^d$ and $g : \mathbb{R}^d \to \mathbb{R}^D$ are the encoder and decoder respectively, and $d < D$. $\|\cdot\|_\ell$ denotes a norm such as the Euclidean norm.

We find that DR methods are very helpful to unsupervised anomaly detection. Specifically, the performance of traditional methods such as kNN performed in the low-dimensional embedding space given by DR methods, e.g. AE+kNN, are much better than their performance in the original high-dimensional data space. Note that our DPAD also reduces the dimensionality of data but it is different from existing DR methods. Existing DR methods aim to preserve the local or global structure of data while our DPAD aims to find a low-dimensional representation with maximum local density. Therefore, the goal of DR in DPAD is consistent with anomaly detection, which means DPAD has the potential to outperform DR+kNN.

DeepSVDD

DeepSVDD (Ruff et al. 2018) aims to enclose the representations of normal data within a hypersphere with minimum volume and solve the following problem

$$\underset{\mathcal{W}}{\text{minimize}} \frac{1}{n}\sum_{i=1}^{n} \|\phi(\mathbf{x}_i;\mathcal{W}) - \mathbf{c}\|^2 + \frac{\lambda}{2}\sum_{l=1}^{L} \left\|\mathbf{W}^l\right\|_F^2$$

where $\mathbf{c}$ is a pre-defined hyper-spherical center, $\mathcal{W} = \{\mathbf{W}^1, \ldots, \mathbf{W}^L\}$ denotes the parameters of layer $l \in \{1, \ldots, L\}$ of neural network $\phi(\mathbf{x};\mathcal{W})$, and λ is a hyperparameter that controls weight decay regularizer. Deep SVDD is able to compress the volume of normal data. This is a global compression and the ideal decision boundary is the hypersphere centered at $\mathbf{c}$. However, in practice, when the dimension of the data is high, the number of data points is small, or the structure of the data is complex, it is difficult to

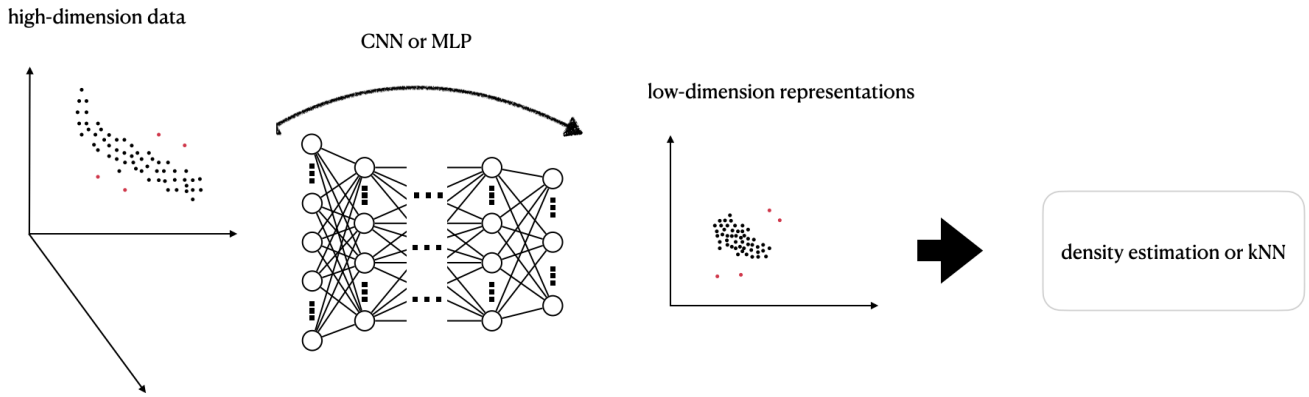


Figure 1: DPAD trains a neural network to learn dense low-dimension representations of training data. The black and red points represent normal data and anomalous data respectively. After training, we can use kNN or other density estimation methods to judge whether a new data point is anomalous or not.

obtain a compact hypersphere, or in other words, it is difficult to include all normal samples into a small hypersphere. In contrast, our DPAD is a local compression method and is able to adapt to data with complex structures.

Proposed Method

Let $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ be a set of D-dimensional training data, in which all or at least most of the samples are normal. Our goal is to learn a model from $\mathcal{D}$ to determine whether a new sample is normal or not. We propose to find a projection $f : \mathbb{R}^D \to \mathbb{R}^d$, where $d < D$, to maximize the density of the data, i.e.,

$$\begin{aligned} &\underset{f}{\text{maximize}}\ \text{density}(\{f(\mathbf{x})\}_{\mathbf{x}\in\mathcal{D}}) \\ &\text{subject to } f \in \mathcal{C}. \end{aligned} \tag{1}$$

In (1), $\mathcal{C}$ is some constraint set to avoid mapping all samples to a single point. Note that estimating the density is computationally expensive. Instead, we replace the density with the local distances between the data points and solve

$$\begin{aligned} &\underset{\mathcal{W}}{\text{minimize}} \sum_{i=1}^{n} \sum_{j\in\mathcal{N}_i} \|f_{\mathcal{W}}(\mathbf{x}_i) - f_{\mathcal{W}}(\mathbf{x}_j)\|^2 \\ &\text{subject to } \mathcal{W} \in \mathcal{C}_{\mathcal{W}} \end{aligned} \tag{2}$$

where $f_{\mathcal{W}}$ is an L-layer neural network parameterized by $\mathcal{W} = \{\mathbf{W}_1, \mathbf{W}_2, \ldots, \mathbf{W}_L\}$, $\mathcal{C}_{\mathcal{W}}$ is some constraint set for the network parameters, and $\mathcal{N}_i$ denotes a local neighborhood of $\mathbf{x}_i$. Nevertheless, determining $\{\mathcal{N}_i\}_{i=1}^n$ still suffers from the curse of dimensionality, is sensitive to noise and outliers, and requires additional efforts or domain knowledge. To tackle these issues, we propose to determine $\{\mathcal{N}_i\}_{i=1}^n$ adaptively and dynamically. Specifically, we solve

$$\begin{aligned} &\underset{\mathcal{W}}{\text{minimize}} \sum_{i=1}^{n} \sum_{j=1}^{n} \|f_{\mathcal{W}}(\mathbf{x}_i) - f_{\mathcal{W}}(\mathbf{x}_j)\|^2 \cdot e_{ij}^{\mathcal{W}} \\ &\text{subject to } \mathcal{W} \in \mathcal{C}_{\mathcal{W}} \end{aligned} \tag{3}$$

where $e_{ij}^{\mathcal{W}} = \exp\left(-\gamma \|f_{\mathcal{W}}(\mathbf{x}_i) - f_{\mathcal{W}}(\mathbf{x}_j)\|^2\right)$ and $\gamma > 0$ is a hyperparameter. The role of $e_{ij}^{\mathcal{W}}$ is explained as follows.

- When the projected samples $f_{\mathcal{W}}(\mathbf{x}_i)$ and $f_{\mathcal{W}}(\mathbf{x}_j)$ are close to each other, $e_{ij}^{\mathcal{W}}$ is close to 1, provided that γ is not too large. Then (3) will make effort on minimizing the distance between $f_{\mathcal{W}}(\mathbf{x}_i)$ and $f_{\mathcal{W}}(\mathbf{x}_j)$.
- When the projected samples $f_{\mathcal{W}}(\mathbf{x}_i)$ and $f_{\mathcal{W}}(\mathbf{x}_j)$ are far away from each other, $e_{ij}^{\mathcal{W}}$ is close to 0, provided that γ is not too small. Then (3) will make less or even no effort on minimizing the distance between $f_{\mathcal{W}}(\mathbf{x}_i)$ and $f_{\mathcal{W}}(\mathbf{x}_j)$.
- The setting of γ is important but not crucial because it can be absorbed into $f_{\mathcal{W}}$ and is thus learned adaptively and implicitly. However, the setting of γ affects the network training because it determines the initial weights $\{e_{ij}^{\mathcal{W}}\}$ once the network parameters are initialized.

Now let's discuss the constraint set $\mathcal{C}_{\mathcal{W}}$. Recall that the constraint is to avoid the case that all projected samples collapse to single points, which lose the original information of the data although the density attains the maximum. A trivial case is that all weights are zero. Therefore, we need to ensure that the norms of the weight matrices are far from zero. Thus, the constraint in (3) is designed as

$$\mathcal{R}(\mathbf{W}_l) \geq \alpha_i, \quad l = 1, 2, \ldots, L, \tag{4}$$

where α_i are positive constants far from zero. For instance, $\mathcal{R}(\mathbf{W}_l)$ can be the Frobenius norm $\|\mathbf{W}_l\|_F$, ℓ_1 norm $\|\mathbf{W}_l\|_1$, or spectral norm $\|\mathbf{W}_l\|_2$. As mentioned in (Yoshida and Miyato 2017), if the weight matrices used in neural networks have large spectral norms, it can cause the neural networks to be sensitive to the perturbation of training data and test data, leading to poor generalization ability. Hence, we may choose $\mathcal{R}(\mathbf{W}_l) = \|\mathbf{W}_l\|_2$, which however is difficult to minimize since its computation is based on singular value decomposition. Note that $\|\mathbf{W}_l\|_2 \leq \|\mathbf{W}_l\|_F$ holds for any $\mathbf{W}_l$. Thus, minimization for $\|\mathbf{W}_l\|_F$, which is much easier, implicitly reduces $\|\mathbf{W}_l\|_2$ and hence improves the generalization ability. To further facilitate the optimization, we use

regularizations instead of constraints on $\mathcal{W}$. Then the final optimization problem is formulated as follows

$$\begin{aligned}\underset{\mathcal{W}}{\text{minimize}} \sum_{i=1}^{n}\sum_{j=1}^{n}\Bigg\{ &\left\|f_{\mathcal{W}}(\mathbf{x}_i)-f_{\mathcal{W}}(\mathbf{x}_j)\right\|^2 \\ &\times \exp\left(-\gamma\left\|f_{\mathcal{W}}(\mathbf{x}_i)-f_{\mathcal{W}}(\mathbf{x}_j)\right\|^2\right)\Bigg\} \qquad (5)\\ &+\lambda\sum_{l=1}^{L}\left|\|\mathbf{W}_l\|_F-1\right|,\end{aligned}$$

where $\lambda > 0$ is a hyperparameter, $f_{\mathcal{W}}(\mathbf{x}) = \mathbf{W}_L(h(\cdots h(\mathbf{W}_2 h(\mathbf{W}_1\mathbf{x}))\cdots))$, and h denotes the activation function. Without loss of generality, we assumed that all activations are the same, for convenience.

Remark 1. *It should be pointed out that in the neural network $f_{\mathcal{W}}$, we cannot include the bias terms. The reason is that unbounded bias terms, which may be learned by the training, can make the activation functions saturated (e.g., sigmoid) or infinite (e.g., ReLU), which further results in model collapse, namely, all data points are mapped to the same point.*

Our dimensionality reduction (DR) method is novel. As shown by (5), it is very different from existing DR methods that aim to compress data with low reconstruction error (e.g., PCA(Jolliffe and Cadima 2016) and autoencoder) or preserve local structures of data (e.g. LLE (Roweis and Saul 2000) and t-SNE (Van der Maaten and Hinton 2008)). Our DR method aims to improve the density (or compactness) of the data in the low-dimensional space, which, shown by the experiments, is useful for anomaly detection.

When the network $f_{\mathcal{W}}$ is well-trained, we can use a density estimation based method such as KDE and LOF (Breunig et al. 2000), to conduct anomaly detection. However, KDE and LOF are time-consuming when n is large and our method with LOF or KDE is not as effective experimentally as it with kNN. Therefore, we propose to use kNN to detect anomalies. To be more precise, given a test sample $\mathbf{x}_{\text{new}}$, we compute

$$\mathbf{z}_{\text{new}} = f_{\mathcal{W}}(\mathbf{x}_{\text{new}}). \qquad (6)$$

For $\mathbf{z}_{\text{new}}$, we find its nearest k neighbors $\{f_{\mathcal{W}}(\mathbf{x}_{\text{new},1}), f_{\mathcal{W}}(\mathbf{x}_{\text{new},2}), \cdots, f_{\mathcal{W}}(\mathbf{x}_{\text{new},k})\} \subseteq \{f_{\mathcal{W}}(\mathbf{x}) : \mathbf{x}\in\mathcal{D}\}$. After that, we compute the average distance from these k neighbors to $\mathbf{z}_{\text{new}}$ and utilize this distance to measure the anomaly of $\mathbf{z}_{\text{new}}$:

$$\text{anomaly score} = \sum_{j=1}^{k}\left\|\mathbf{z}_{\text{new}} - f_{\mathcal{W}}(\mathbf{x}_{\text{new},j})\right\|^2. \qquad (7)$$

In general, we train a neural network to learn dense representations of normal data (1) with our objective function (5). As for test data, we utilize the trained neural network to generate a representation of the test data (6). Subsequently, we find its nearest K representations generated by training data , and calculate the sum of distance from test representation to its nearest K neighbors as an anomaly score (7). For convenience, we call our method (5) Dense Projection based Anomaly Detection (DPAD).

Optimization

Training Settings

In the training stage, to ensure that the distance between any two representations of training data is fully considered and optimized, we refrain from using mini-batch which may lead the model to repeatedly consider the distance between representations generated by training data in the same batch, thereby overlooking the distances between representations of the training data from different batches which may be more similar to each other. Moreover, the setting of hyperparameter γ controls the initialization of weights for $e_{ij}^{\mathcal{W}}$ and thus determines whether the model will shrink the distance between $f_{\mathcal{W}}(\mathbf{x}_i)$ and $f_{\mathcal{W}}(\mathbf{x}_j)$ at the beginning of training. An excessively large value of γ would lead the model to attempt increasing the distances between all points to minimize the objective function, as we observe that the objective function decreases with increasing distance $||f_{\mathcal{W}}(\mathbf{x}_i) - f_{\mathcal{W}}(\mathbf{x}_j)||$ when $||f_{\mathcal{W}}(\mathbf{x}_i) - f_{\mathcal{W}}(\mathbf{x}_j)|| \geq 1/\gamma$. To handle this problem, $e_{ij}^{\mathcal{W}}$ is excluded from the backpropagation process so it will be only a parameter or weight of distance and we set γ to a relatively small numerical value. The optimization details are presented in Algorithm 1.

Algorithm 1: Training and testing processes of DPAD

Input: $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, m, $\gamma \geq 0, \lambda \geq 0, k \geq 1$
Training stage of DPAD:
  for $B = 1, \ldots, m$ **do**
    $e_{ij}^{\mathcal{W}} = \exp\left(-\gamma\left\|f_{\mathcal{W}}(\mathbf{x}_i) - f_{\mathcal{W}}(\mathbf{x}_j)\right\|^2\right)$.detach()
    Dist sum$= \sum_{i=1}^{n}\sum_{j=1}^{n}\left(\left\|f_{\mathcal{W}}(\mathbf{x}_i) - f_{\mathcal{W}}(\mathbf{x}_j)\right\|^2 e_{ij}^{\mathcal{W}}\right)$
    Loss$=$ Dist sum$+\lambda\sum_{l=1}^{L}\left|\|\mathbf{W}_l\|_F - 1\right|$
    $\mathcal{W} = \mathcal{W}-$ Gradient-Step(Loss)
  end for
Testing stage of DPAD:
  Input test data $\mathbf{x}_{new}$
  Compute $\mathbf{z}_{\text{new}} = f_{\mathcal{W}}(\mathbf{x}_{\text{new}})$
  Find the nearest k neighbors of $\mathbf{z}_{\text{new}}$ from $\{f_{\mathcal{W}}(\mathbf{x}) : \mathbf{x}\in\mathcal{D}\}$:
    $\{f_{\mathcal{W}}(\mathbf{x}_{\text{new},1}), f_{\mathcal{W}}(\mathbf{x}_{\text{new},2}), \cdots, f_{\mathcal{W}}(\mathbf{x}_{\text{new},k})\}$
  Anomaly Score $= \sum_{j=1}^{k}\left\|\mathbf{z}_{\text{new}} - f_{\mathcal{W}}(\mathbf{x}_{\text{new},j})\right\|^2$

Space and Time Complexity

Suppose $\mathbf{W}_l \in \mathbb{R}^{d_l\times d_{l-1}}$, $l = 1, 2, \ldots, L$, and consider a mini-batch of b samples, where $d_L = d$ and $d_0 = D$. The time complexity per iteration (including the forward and backward propagations) is $\mathcal{O}(b\sum_{l=1}^{L} d_{l-1}d_l)$ and the space complexity is $\mathcal{O}(b\sum_{l=1}^{L+1} d_{l-1} + \sum_{l=1}^{L} d_{l-1}d_l)$. In the testing stage, for a test sample, the time complexity is $\mathcal{O}(\sum_{l=1}^{L} d_{l-1}d_l + dn)$, in which the first part is from the computation of $f_{\mathcal{W}}(\mathbf{x}_{new})$ and the second part is from kNN. In sum, the time and space complexities of the proposed method DPAD are both linear with the number of training data. Therefore, DPAD can be applied to large datasets.

Theoretical Analysis

First we provide a Lipschitz constant τ_f for $f_{\mathcal{W}}$.

Lemma 1. *Given the neural network* $f_{\mathcal{W}}(\mathbf{x}) = \mathbf{W}_L(h(\cdots h(\mathbf{W}_2 h(\mathbf{W}_1\mathbf{x}))\cdots))$*, denote* ρ *the Lipschitz constant of* h *and suppose* $\|\mathbf{W}_l\|_2 \leq \beta_l$*. Let* $\tau_f = \rho^{L-1}\prod_{l=1}^{L}\beta_l$*. Then for any* $\mathbf{x}_1$ *and* $\mathbf{x}_2$*, the following inequality holds*

$$\|f_{\mathcal{W}}(\mathbf{x}_1) - f_{\mathcal{W}}(\mathbf{x}_2)\| \leq \tau_f\|\mathbf{x}_1 - \mathbf{x}_2\|. \tag{8}$$

The lemma, proved in shows the sensitivity of $f_{\mathcal{W}}$ to the distances between any two data points in $\mathcal{D}$. The following lemma shows the upper bound of the spectral norm of a random Gaussian matrix.

Lemma 2. *(Bandeira and Van Handel 2016) Given an* $d \times d$ *random Gaussian matrix* $\mathbf{N}$ *with* $N_{ij} \sim \mathcal{N}(0, \sigma_{ij}^2)$*, the following inequality holds*

$$\|\mathbf{N}\|_2 \leq \max_i \sqrt{\sum_j \sigma_{ij}^2} + \max_{ij}|\sigma_{ij}|\sqrt{\log d} \tag{9}$$

Based on Lemma 1 and Lemma 2, we have the following theorem (proved in the appendix), which provides a lower bound for the weight $e_{ij}^{\mathcal{W}}$ at the random initialization stage of the $f_{\mathcal{W}}$.

Theorem 1. *Let* $\mathcal{W}(0)$ *be the initialized parameters drawn from* $\mathcal{N}(0, \sigma^2)$*. Denote* $d_l \times d_{l-1}$ *the shape of* $\mathbf{W}_l$ *and let* $\bar{d}_l = \max(d_l, d_{l-1})$*,* $l = 1, 2, \ldots, L$*. Then the following inequality holds:*

$$\begin{aligned} e_{ij}^{\mathcal{W}(0)} \geq & \exp\Bigg(- \gamma\rho^{(2L-2)}\sigma^{2L}\|\mathbf{x}_i - \mathbf{x}_j\|^2 \\ & \times \prod_{l=1}^{L}(\sqrt{\bar{d}_l} + \sqrt{\log \bar{d}_l})^2\Bigg). \end{aligned} \tag{10}$$

The theorem indicates that the initialized $f_{\mathcal{W}}$ is able to preserve the local similarity of the original data in $\mathcal{D}$ provided that the network is not too complex. Therefore, the problem that the network reduces the distance of representations of dissimilar data at the beginning of training will not occur.

Experiments

Datasets and Baselines

We choose CIFAR-10 (Krizhevsky, Hinton et al. 2009) and Fashion-MNIST (Xiao, Rasul, and Vollgraf 2017) as our image datasets, Arrhythmia (Rayana 2016), Abalone (Dua, Graff et al. 2017), Campaign (Han et al. 2022), and MAGIC Gama (Han et al. 2022) as our tabular datasets to test the proposed method DPAD. The statistics about the datasets are in Table 1.

We compare DPAD with classical methods, dimensionality reduction methods followed by kNN, and state-of-the-art AD methods. It is noteworthy that we also used LOF as a detection method following DR, but the performance was much worse than kNN.

Datasets	# Samples	# Features	# Classes
CIFAR-10	60000	$32 \times 32 \times 3$	10
Fashion-MNIST	70000	28×28	10
Arrhythmia	452	274	2
Abalone	1920	8	2
Campaign	41188	62	2
MAGIC Gama	19020	10	2

Table 1: Statistics of the datasets

- **Classical methods**: kNN, k-Means (MacQueen et al. 1967), LOF (Breunig et al. 2000), OCSVM (Schölkopf et al. 2001), isolation forest (IF) (Liu, Ting, and Zhou 2008), KDE (Parzen 1962), and DAE (Vincent et al. 2008).
- **Dimensionality reduction methods**: PCA (Jolliffe and Cadima 2016), t-SNE (Van der Maaten and Hinton 2008), and UMAP (McInnes, Healy, and Melville 2018).
- **State-of-the-art methods**: E2E-AE and DAGMM (Zong et al. 2018), DCN (Caron et al. 2018), ADGAN (Deecke et al. 2019), DSVDD (Ruff et al. 2018), OCGAN (Perera, Nallapati, and Xiang 2019), TQM (Wang, Sun, and Yu 2019), GOAD (Bergman and Hoshen 2020), DROCC (Goyal et al. 2020), HRN (Hu et al. 2020), SCADN (Yan et al. 2021), NeuTraL AD (Qiu et al. 2021), GOCC (Shenkar and Wolf 2021), and IGD (Chen et al. 2022).

Implementation and Evaluation Details

In this section, we introduce experimental settings and describe the implementation details of the proposed method. For the two mentioned image datasets, we use Le-Net-based CNN as our basic network structure. We conduct 10 one-class classification tasks, choosing one of the 10 classes as the normal class every time. To further evaluate the performance of our method, we conducted an additional set of challenging experiments, where we selected 9 out of the 10 classes as the normal classes for training, while the testing samples remained the same as before. For the compared methods in the experiment, we obtain their performance directly from their paper except for k-Means, DROCC, and DR+kNN methods for which we run the officially released code or our code respectively to obtain the results. We run the proposed methods 5 times with 100 epochs optimization to get the final average result. To maintain consistency with previous methods, we use the AUC metric to evaluate the performance on image datasets and use the F1 score to evaluate the performance on tabular datasets.

Results on Image Datasets

Table 2 and Table 3 provide a summary and comparison of our method with other methods in terms of their AUC performance on every class of CIFAR-10 and Fashion-MNIST datasets. Based on the performance, we draw the following observations:

- In comparison with classical methods like OCSVM and IF, our approach consistently achieves higher AUC

Normal class	Airplane	Auto-mobile	Bird	Cat	Deer	Dog	Frog	Horse	Ship	Truck
(no DR) kNN	91.2	98.6	88.5	93.6	89.4	90.0	81.7	98.4	88.5	96.5
(no DR) k-Means	90.3	98.6	88.5	93.8	88.1	90.5	82.4	98.1	89.8	97.0
(no DR) LOF (Breunig et al. 2000)	66.6	45.3	64.1	51.6	67.5	51.7	67.7	52.9	69.3	41.6
OCSVM (Schölkopf et al. 2001)	61.6	63.8	50.0	55.9	66.0	62.4	74.7	62.6	74.9	75.9
KDE (Parzen 1962)	61.2	64.0	50.1	56.4	66.2	62.4	74.9	62.6	75.1	76.0
IF (Liu, Ting, and Zhou 2008)	66.1	43.7	64.3	50.5	**74.3**	52.3	70.7	53.0	69.1	53.2
DAE (Vincent et al. 2008)	41.1	47.8	61.6	56.2	72.8	51.3	68.8	49.7	48.7	37.8
DAGMM (Zong et al. 2018)	41.4	57.1	53.8	51.2	52.2	49.3	64.9	55.3	51.9	54.2
ADGAN (Deecke et al. 2019)	63.2	52.9	58.0	60.6	60.7	65.9	61.1	63.0	74.4	64.2
DSVDD (Ruff et al. 2018)	61.7	65.9	50.8	59.1	60.9	65.7	67.7	67.3	75.9	73.1
OCGAN (Perera, Nallapati, and Xiang 2019)	75.7	53.1	64.0	62.0	72.3	62.0	72.3	57.5	**82.0**	55.4
TQM (Wang, Sun, and Yu 2019)	40.7	53.1	41.7	58.2	39.2	62.6	55.1	63.1	48.6	58.7
DROCC* (Goyal et al. 2020)	**79.2**	**74.9**	**68.3**	62.3	70.3	66.1	68.1	**71.3**	62.3	**76.6**
HRN (Hu et al. 2020)	77.3	69.9	60.6	64.4	71.5	67.4	**77.4**	64.9	**82.5**	**77.3**
AE+kNN*	77.7	62.7	59.5	57.6	65.3	58.3	75.5	62.8	79.7	66.4
PCA+kNN*	68.7	44.7	68.1	51.0	77.0	49.6	73.4	51.3	69.0	43.7
t-SNE+kNN*	**78.4**	72.1	68.3	**66.7**	70.3	**68.8**	75.5	70.3	82.0	72.6
UMAP+kNN*	75.6	66.7	63.0	60.1	64.9	64.0	73.4	63.8	77.9	67.2
DPAD	78.0 (0.3)	**75.0 (0.2)**	**68.1 (0.5)**	**66.7 (0.4)**	**77.9 (0.8)**	**68.6 (0.3)**	**81.2 (0.4)**	**74.8 (0.2)**	79.1 (1.0)	76.1 (0.2)

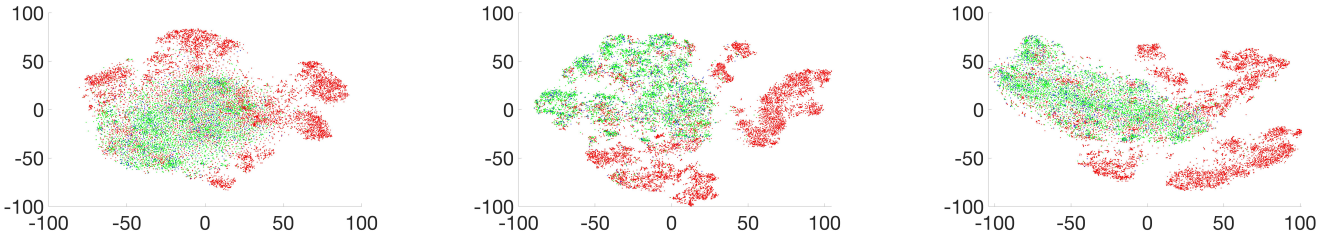
Table 2: Average AUC(%) of one-class anomaly detection on CIFAR-10. * means we reproduced the results using the officially released code. The best two results are marked in bold.

Normal class	T-shirt	Trouser	Pullover	Dress	Coat	Sandal	Shirt	Sneaker	Bag	Ankle-boot
(no DR) kNN	91.2	98.6	88.5	93.6	89.4	90.0	81.7	98.4	88.5	96.5
(no DR) k-Means	90.3	98.6	88.5	93.8	88.1	90.5	82.4	98.1	89.8	97.0
(no DR)LOF (Breunig et al. 2000)	80.6	94.6	82.4	88.6	91.0	88.6	78.6	96.4	75.8	97.4
OCSVM (Schölkopf et al. 2001)	86.1	93.9	85.6	85.9	84.6	81.3	78.6	97.6	79.5	97.8
KDE (Parzen 1962)	68.7	91.0	86.0	91.9	84.6	88.5	58.7	94.1	69.3	90.1
IF (Liu, Ting, and Zhou 2008)	91.0	97.8	87.2	93.2	90.5	93.0	80.2	98.2	88.7	95.4
DAE (Vincent et al. 2008)	86.7	97.8	80.8	91.4	86.5	92.1	73.8	97.7	78.2	96.3
DAGMM (Zong et al. 2018)	42.1	55.1	50.4	57.0	26.9	70.5	48.3	83.5	49.9	34.0
ADGAN (Deecke et al. 2019)	89.9	81.9	87.6	91.2	86.5	89.6	74.3	97.2	89.0	97.1
DSVDD (Ruff et al. 2018)	79.1	94.0	83.0	82.9	87.0	80.3	74.9	94.2	79.1	93.2
OCGAN (Perera, Nallapati, and Xiang 2019)	85.5	93.4	85.0	88.1	85.8	88.5	77.5	93.9	82.7	97.8
TQM (Wang, Sun, and Yu 2019)	92.2	95.8	89.9	93.0	92.2	89.4	84.4	98.0	94.5	**98.3**
DROCC* (Goyal et al. 2020)	88.1	97.7	87.6	87.7	87.2	91.0	77.1	95.3	82.7	95.9
HRN (Hu et al. 2020)	92.7	98.5	88.5	93.1	92.1	91.3	79.8	**99.0**	94.6	**98.8**
AE+KNN*	86.9	98.4	78.9	93.3	83.1	92.2	79.3	98.4	86.5	94.5
PCA+kNN*	92.8	**99.0**	90.0	95.4	91.1	92.6	**85.1**	98.7	91.3	96.9
t-SNE+kNN*	**95.2**	98.3	92.2	**97.1**	91.6	**98.0**	84.1	96.7	**98.0**	97.9
UMAP+kNN*	**94.3**	98.0	**92.1**	**96.9**	**92.5**	**97.4**	**85.6**	97.3	**98.8**	98.2
DPAD	93.7 (0.2)	**98.7 (0.0)**	**90.3 (0.0)**	94.7 (0.3)	**92.2 (0.1)**	93.9 (0.8)	82.3 (0.1)	**98.7 (0.1)**	94.2 (0.6)	98.1 (0.2)

Table 3: Average AUC(%) of one-class anomaly detection on Fashion-MNIST. * means we reproduced the results using the officially released code. The best two results are marked in bold.

scores for all classes in both two datasets. An interesting phenomenon is that IF outperforms all other deep methods in some classes except for DPAD.

- For DR methods, UMAP+kNN outperforms most methods in most classes in Fashion-MNIST, and DR methods are excellent when handling data with a simple structure



(a) DeepSVDD
AUC : 83.0

(b) AE+KNN
AUC : 78.9

(c) DPAD
AUC : 90.3

Figure 2: t-SNE visualization of the learned embedding space of "Pullover" class of Fashion-MNIST. Note that points marked in green, blue, red correspond to training data, test normal data, and test anomalous data respectively.

like Fashion-MNIST. But it's worth noting that among all other methods, DPAD has the smallest gap with DR methods in most classes. When it comes to complex data like CIFAR-10, DPAD outperforms all DR methods with significant differences.

- As for deep learning based methods, DPAD outperforms several methods such as DSVDD and OCGAN in all classes and gets the highest two scores in most classes. Although DPAD doesn't achieve the top 2 best performances in some specific classes, as shown in Table 4, in terms of the average performance over all classes, our method is the best state-of-the-art method. In contrast to DSVDD and DROCC which assume normal samples in the embedding space lie in a hyper-sphere, our method does not make assumptions about specific shapes formed by training data which is capable of yielding better performance in cases of complex data structures.

We employ t-SNE (Van der Maaten and Hinton 2008) to visualize the representations formed by the neural network in our method and DSVDD, and encoder in AE+kNN. Specifically, We visualize the training data, normal test data, and anomalous test data with different colors. Figure 2 shows the visualization of the class "Pullover" in Fashion-MNIST. From this figure, we have the following observations.

- First, our method indeed compacts the training data in the embedding space, and there is not only a significant overlap between the normal test data and the training data but also a clear separation between training data and anomalous test data. We conclude that our method obtains a clear decision boundary to distinguish normal data and anomalous data.
- Second, Compared to DSVDD and AE, our method can learn a better decision boundary to distinguish normal data and abnormal data, which is consistent with the mentioned experimental results.

Table 4 shows the average performance on CIFAR-10 and Fashion-MNIST over all 10 classes. The two latest methods SCADN and IGD (Scratch) are also compared in the table, though their performance on each single class was not reported in their papers. From the table, we draw the following observation:

Datasets	CIFAR-10	F-MNIST
(no DR) kNN*	59.5	91.6
(no DR) k-Means*	62.0	91.7
(no DR) LOF (Breunig et al. 2000)	57.8	87.4
OCSVM (Schölkopf et al. 2001)	64.7	87.0
IF (Liu, Ting, and Zhou 2008)	59.7	91.5
KDE (Parzen 1962)	64.9	82.3
DAE (Vincent et al. 2008)	53.5	88.1
DAGMM (Zong et al. 2018)	53.1	51.7
ADGAN (Deecke et al. 2019)	62.4	88.4
DSVDD (Ruff et al. 2018)	64.8	84.7
OCGAN (Perera et al. 2019)	65.6	87.8
TQM (Wang, Sun, and Yu 2019)	52.1	92.7
DROCC* (Goyal et al. 2020)	69.9	89.0
HRN (Hu et al. 2020)	71.3	92.8
SCADN (Yan et al. 2021)	66.9	—
IGD (Chen et al. 2022)	**74.3**	92.0
AE+kNN*	65.2	89.1
PCA+kNN*	58.7	93.3
t-SNE+kNN*	72.3	**94.9**
UMAP+kNN*	67.7	**95.2**
DPAD	**74.5**	93.7

Table 4: Average AUCs(%) over all 10 classes on CIFAR-10 and Fashion-MNIST. Note that the best two results are marked in bold.

- On Fashion-MNIST, classical methods and dimensionality reduction methods demonstrate excellent performance, with UMAP+kNN surpassing all state-of-the-art methods. We attribute this phenomenon to the comparatively simple data structure of Fashion-MNIST. Despite DPAD not achieving optimal performance, it remains the state-of-the-art method whose performance is closest to that of UMAP+kNN.
- On CIFAR-10, due to its complex data structure, SOTA methods demonstrate superior performance compared to classical methods and dimensionality reduction methods. DPAD outperforms other methods, which verified its effectiveness in handling data with high complexity.

Datasets	Abalone	Arrhythmia	Campaign	MAGIC-gamma
(no DR) kNN*	61.5± 0.0	**63.8± 0.0**	72.1± 0.0	75.2± 0.0
(no DR) k-Means*	61.8± 0.0	62.8± 0.0	72.0± 0.0	70.6± 0.0
(no DR) LOF* (Breunig et al. 2000)	33.0 ± 1.0	51.0 ± 1.0	64.0± 0.0	68.0± 0.0
OCSVM* (Schölkopf et al. 2001)	48.0 ± 0.0	46.0 ± 0.0	67.0± 0.0	67.0± 0.0
E2E-AE (Zong et al. 2018)	33.0 ± 3.0	45.0 ± 3.0	-	-
DCN (Caron et al. 2018)	40.0 ± 1.0	38.0 ± 3.0	-	-
DAGMM (Zong et al. 2018)	20.0 ± 3.0	49.0 ± 3.0	-	-
DSVDD (Ruff et al. 2018)	62.0 ± 1.0	54.0 ± 1.0	61.7 ± 6.4*	65.5 ± 0.3*
DROCC* (Goyal et al. 2020)	**68.0 ± 2.0**	32.3 ± 1.8	65.5 ± 0.9	58.0 ± 1.4
GOAD (Bergman and Hoshen 2020)	61.0 ± 2.0	52.0 ± 2.3	64.5 ± 0.7*	61.6 ± 0.1*
NeuTraL AD* (Qiu et al. 2021)	62.1 ± 2.8	60.3 ± 1.1	63.2 ± 8.0	69.6 ± 2.8
GOCC* (Shenkar and Wolf 2021)	66.1 ± 4.3	61.8 ± 1.8	**74.1 ± 2.5**	66.7 ± 0.4
PCA+kNN*	56.7± 0.0	25.0± 0.0	67.2± 0.0	72.9± 0.0
t-SNE+kNN*	61.9± 0.0	13.7± 0.0	67.4± 0.0	**76.6± 0.0**
UMAP+kNN*	61.7± 0.0	11.4± 0.0	66.9± 0.0	**74.8± 0.0**
DPAD	**66.7 ± 1.5**	**66.7 ± 0.0**	**73.4 ± 1.5**	74.0 ± 0.5

Table 5: Average F1-scores(%) with the standard deviation of each method on four tabular datasets. * means we reproduced the results using the officially released code. The best two results are marked in bold.

Results on Tabular Datasets

In Table 5, we summarize the F1-scores of all methods on four tabular datasets. It can be observed that DPAD significantly outperforms several baseline methods such as OCSVM, DCN, and DAGMM. Note that for Campaign and MAGIC-gamma, we run the officially released code or our own code to get the results. When faced with low-dimensional data such as Campaign and MAGIC-gamma, classical methods and DR methods can even get better results than some deep learning based methods such as DSVDD and DROCC. Compared with methods designed for tabular data such as NeuTraL AD and GOCC, our DPAD is more effective. Moreover, Arrhythmia is a more challenging dataset with fewer samples and more attributes, and DPAD exhibits a performance improvement of 4% over the second-best method while the performance of DR methods is the worst indicating they fail when faced with complex datasets.

Experiment with Multi-Class Normality

In real anomaly detection scenarios, the normal data may consist of multiple classes with small associations. To evaluate the performance of our method under such practical conditions, we conduct experiments on Fashion-MNIST and CIFAR-10 datasets by selecting one class as an anomalous class and the remaining nine classes as normal classes. Therefore, we conducted 10 experiments for each dataset. In this setup, the normal samples come from different classes and have relatively small associations, making it a more challenging task than traditional one-class classification. We compare our method with OCSVM, SVDD, DROCC, HRN, and dimensionality reduction methods.

Table 6 shows the average performance. We have the following observations:

- Compared to traditional one-class classification tasks, all methods experience a significant decrease in average AUC which demonstrates that the 9-1 experiments are indeed more challenging than the 1-9 experiments shown in previous tables.

- Although dimensionality reduction methods perform well on Fashion-MNIST, their average AUCs are around 50 on CIFAR-10, indicating they failed to handle data with complex structures. DPAD achieves the best performance on CIFAR-10, indicating it is more effective on anomaly detection in complex real scenarios than other state-of-the-art methods. Its success mainly stems from the ability to learn a decision boundary locally without any assumption on the shape of the decision boundary.

Datasets	CIFAR-10	F-MNIST
(no DR) kNN	52.1	71.6
(no DR) k-Means	48.8	68.8
(no DR) LOF(Breunig et al. 2000)	50.0	50.0
OCSVM (Schölkopf et al. 2001)	49.0	57.2
DSVDD (Ruff et al. 2018)	52.3	65.9
DROCC (Goyal et al. 2020)	54.3	54.8
HRN (Hu et al. 2020)	50.3	41.1
PCA+kNN	52.2	74.8
t-SNE+kNN	51.3	**78.7**
UMAP+kNN	51.2	74.4
AE+kNN	51.4	69.0
DPAD	**66.1**	70.2

Table 6: Average AUCs(%) of 9-1 experiments on CIFAR-10 and Fashion-MNIST. Note that we run the officially released code to get results and the best result is marked in bold.

Ablation Study

We study the contributions of the two components of our method. Table 7 gives the ablation results on Fashion-MNIST and CIFAR-10. We can see that both $e_{ij}^{\mathcal{W}}$ and $\mathcal{C}_{\mathcal{W}}$ are necessary.

Datasets	CIFAR-10	Fashion-MNIST
DPAD without $e_{ij}^{\mathcal{W}}$ and $\mathcal{C}_{\mathcal{W}}$	57.8	87.0
DPAD without $e_{ij}^{\mathcal{W}}$	68.5	91.3
DPAD without $\mathcal{C}_{\mathcal{W}}$	68.5	89.2
DPAD	**74.5**	**93.7**

Table 7: Average AUCs(%) of different components of DPAD on the image datasets.

To show that the performance is not significantly dependent on values of γ and λ, we choose different values of them to see the difference in one-class classification experiments on Fashion-MNIST. Table 8 shows the results, where the differences are tiny if γ and λ are in some reasonable ranges respectively. Nevertheless, substantial performance degradation is evident when γ is 100, corroborating that an excessively large γ inhibits the learning of dense representations, and then impacts performance. Besides, results show that our method is not sensitive to the value of λ.

Datasets		Fashion-MNIST
$\gamma = 0.001$		92.3
$\gamma = 0.01$		**93.3**
$\gamma = 0.1$		91.3
$\gamma = 1$	$\lambda = 1$	90.5
$\gamma = 10$		91.4
$\gamma = 100$		89.2
	$\lambda = 0$	89.2
	$\lambda = 0.01$	92.5
	$\lambda = 0.1$	92.5
$\gamma = 0.01$	$\lambda = 10$	**92.6**
	$\lambda = 100$	91.9
	$\lambda = 1000$	92.0

Table 8: Average AUCs(%) of different values of hyper-parameter γ and λ on Fashion-MNIST.

Conclusions

We have presented a novel and simple method DPAD for unsupervised anomaly detection. The main idea is to learn dense representations of normal data using neural networks and detect anomalous data based on its local density. Compared with other methods, DPAD does not rely on any assumption on the shape of normal data and the decision boundary formed by representations of normal data and only tries to gather representations of similar normal data. For this reason, DPAD is not only effective on classical one-class classification tasks but also outperforms other methods when normal data consists of multiple classes with small associations. Our experimental results demonstrate that DPAD is as effective as state-of-the-art AD methods on both image and tabular datasets and has significant improvements in a few cases.

Proof of Theoretical Results

Proof for Lemma 1

Given the architecture of $f_{\mathcal{W}}$, we have

$$\begin{aligned}
&\|f_{\mathcal{W}}(\mathbf{x}_1) - f_{\mathcal{W}}(\mathbf{x}_2)\| \\
=&\|\mathbf{W}_L(h(\cdots h(\mathbf{W}_2 h(\mathbf{W}_1\mathbf{x}_1))\cdots)) \\
&- \mathbf{W}_L(h(\cdots h(\mathbf{W}_2 h(\mathbf{W}_1\mathbf{x}_2))\cdots))\| \\
\leq&\|\mathbf{W}_L\|_2\|h(\cdots h(\mathbf{W}_2 h(\mathbf{W}_1\mathbf{x}_1))\cdots) \\
&- h(\cdots h(\mathbf{W}_2 h(\mathbf{W}_1\mathbf{x}_2))\cdots)\| \\
\leq&\rho\|\mathbf{W}_L\|_2\|\cdots h(\mathbf{W}_2 h(\mathbf{W}_1\mathbf{x}_1))\cdots \\
&- h(\cdots \mathbf{W}_2 h(\mathbf{W}_1\mathbf{x}_2))\cdots\| \\
&\vdots \\
\leq&\rho^{L-1}\Big(\prod_{l=1}^{L}\|\mathbf{W}_l\|_2\Big)\|\mathbf{x}_1 - \mathbf{x}_2\| \\
\leq&\rho^{L-1}\Big(\prod_{l=1}^{L}\beta_l\Big)\|\mathbf{x}_1 - \mathbf{x}_2\| \\
=&\tau_f\|\mathbf{x}_1 - \mathbf{x}_2\|.
\end{aligned} \tag{11}$$

Proof for Theorem 1

For our $f_{\mathcal{W}}$, the weight matrices $\mathbf{W} \in \mathbb{R}^{d_l \times d_{l-1}}$ are initialized from $\mathcal{N}(0, \sigma^2)$. According to Lemma 2, we have

$$\|\mathbf{W}_l(0)\|_2 \leq \sqrt{\bar{d}_l}\sigma + \sqrt{\log \bar{d}_l}\sigma \tag{12}$$

where $\bar{d}_l = \max(d_l, d_{l-1})$. The inequation shows an upper bound of the spectral norm of $\mathbf{W}_l$ when it is initialized by a Gaussian distribution with variance σ^2. Now for Lemma 1, we have

$$\tau_f = \rho^{L-1}\prod_{l=1}^{L}(\sqrt{\bar{d}_l}\sigma + \sqrt{\log \bar{d}_l}\sigma). \tag{13}$$

It follows from Lemma 1 that

$$\begin{aligned}
\|f_{\mathcal{W}}(\mathbf{x}_i) - f_{\mathcal{W}}(\mathbf{x}_j)\| \leq \rho^{L-1}\sigma^L\|\mathbf{x}_i - \mathbf{x}_j\| \\
\times\prod_{l=1}^{L}(\sqrt{\bar{d}_l} + \sqrt{\log \bar{d}_l}).
\end{aligned} \tag{14}$$

Thus we can get an upper bound for any $e_{ij}^{\mathcal{W}(0)}$:

$$\begin{aligned}
e_{ij}^{\mathcal{W}(0)} &= \exp\left(-\gamma\|f_{\mathcal{W}}(\mathbf{x}_i) - f_{\mathcal{W}}(\mathbf{x}_j)\|^2\right) \\
&\geq \exp\Bigg(-\gamma\rho^{2L-2}\sigma^{2L}\|\mathbf{x}_i - \mathbf{x}_j\|^2 \\
&\quad\times\prod_{l=1}^{L}(\sqrt{\bar{d}_l} + \sqrt{\log \bar{d}_l})^2\Bigg).
\end{aligned} \tag{15}$$

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China under Grants No.62106211 and No.62072151, the General Program JCYJ20210324130208022 of Shenzhen Fundamental Research, the research funding T00120210002 of Shenzhen Research Institute of Big Data, the Guangdong Key Lab of Mathematical Foundations for Artificial Intelligence, Anhui Provincial Natural Science Fund for the Distinguished Young Scholars (2008085J30), Open Foundation of Yunnan Key Laboratory of Software Engineering (2023SE103), CCF-Baidu Open Fund and CAAI-Huawei MindSpore Open Fund, and the funding UDF01001770 of The Chinese University of Hong Kong, Shenzhen.

References

Bandeira, A. S.; and Van Handel, R. 2016. Sharp nonasymptotic bounds on the norm of random matrices with independent entries.

Bergman, L.; and Hoshen, Y. 2020. Classification-based anomaly detection for general data. *arXiv preprint arXiv:2005.02359*.

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. In Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; and Oh, A., eds., *Advances in Neural Information Processing Systems*, volume 35, 14317–14330. Curran Associates, Inc.

Caron, M.; Bojanowski, P.; Joulin, A.; and Douze, M. 2018. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European conference on computer vision (ECCV)*, 132–149.

Chandola, V.; Banerjee, A.; and Kumar, V. 2009. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3): 1–58.

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.

Deecke, L.; Vandermeulen, R.; Ruff, L.; Mandt, S.; and Kloft, M. 2019. Image anomaly detection with generative adversarial networks. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2018, Dublin, Ireland, September 10–14, 2018, Proceedings, Part I 18*, 3–17. Springer.

Dua, D.; Graff, C.; et al. 2017. UCI machine learning repository.

Fan, J.; Chow, T. W.; Zhao, M.; and Ho, J. K. 2018. Nonlinear dimensionality reduction for data with disconnected neighborhood graph. *Neural Processing Letters*, 47: 697–716.

Fan, J.; Wang, W.; and Zhang, H. 2017. AutoEncoder based high-dimensional data fault detection system. In *2017 IEEE 15th International Conference on Industrial Informatics (INDIN)*, 1001–1006.

Fan, J.; and Wang, Y. 2014. Fault detection and diagnosis of non-linear non-Gaussian dynamic processes using kernel dynamic independent component analysis. *Information Sciences*, 259: 369–379.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. *Advances in neural information processing systems*, 27.

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.

Hinton, G. E.; and Salakhutdinov, R. R. 2006. Reducing the dimensionality of data with neural networks. *science*, 313(5786): 504–507.

Hu, W.; Wang, M.; Qin, Q.; Ma, J.; and Liu, B. 2020. HRN: A holistic approach to one class learning. *Advances in neural information processing systems*, 33: 19111–19124.

Jolliffe, I. T.; and Cadima, J. 2016. Principal component analysis: a review and recent developments. *Philosophical transactions of the royal society A: Mathematical, Physical and Engineering Sciences*, 374(2065): 20150202.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images.

LeCun, Y.; Bengio, Y.; and Hinton, G. 2015. Deep learning. *nature*, 521(7553): 436–444.

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.

MacQueen, J.; et al. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, 281–297. Oakland, CA, USA.

McInnes, L.; Healy, J.; and Melville, J. 2018. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*.

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.

Parzen, E. 1962. On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3): 1065–1076.

Perera, P.; Nallapati, R.; and Xiang, B. 2019. Ocgan: One-class novelty detection using gans with constrained latent representations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2898–2906.

Pidhorskyi, S.; Almohsen, R.; and Doretto, G. 2018. Generative probabilistic novelty detection with adversarial autoencoders. *Advances in neural information processing systems*, 31.

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.

Rayana, S. 2016. ODDS Library [http://odds. cs. stonybrook. edu]. *Stony Brook University, Department of Computer Science, Stony Brook, NY*.

Roweis, S. T.; and Saul, L. K. 2000. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500): 2323–2326.

Ruff, L.; Kauffmann, J. R.; Vandermeulen, R. A.; Montavon, G.; Samek, W.; Kloft, M.; Dietterich, T. G.; and Müller, K.-R. 2021. A unifying review of deep and shallow anomaly detection. *Proceedings of the IEEE*, 109(5): 756–795.

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.

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. 2021. Anomaly detection for tabular data with internal contrastive learning. In *International Conference on Learning Representations*.

Sun, Y.; Han, Y.; and Fan, J. 2023. Laplacian-Based Cluster-Contractive t-SNE for High-Dimensional Data Visualization. *ACM Trans. Knowl. Discov. Data*, 18(1).

Sun, Y.; Ming, Y.; Zhu, X.; and Li, Y. 2022. Out-of-distribution detection with deep nearest neighbors. In *International Conference on Machine Learning*, 20827–20840. PMLR.

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

Tenenbaum, J. B.; Silva, V. d.; and Langford, J. C. 2000. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500): 2319–2323.

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

Vincent, P.; Larochelle, H.; Bengio, Y.; and Manzagol, P.-A. 2008. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, 1096–1103.

Wang, J.; Sun, S.; and Yu, Y. 2019. Multivariate triangular quantile maps for novelty detection. *Advances in Neural Information Processing Systems*, 32.

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.; Sun, R.; and Fan, J. 2023. Restricted Generative Projection for One-Class Classification and Anomaly Detection. *arXiv preprint arXiv:2307.04097*.

Xiao, H.; Rasul, K.; and Vollgraf, R. 2017. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*.

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.

Yoshida, Y.; and Miyato, T. 2017. Spectral norm regularization for improving the generalizability of deep learning. *arXiv preprint arXiv:1705.10941*.

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*.