A Non-convex Anti-noise Model for Anomaly Detection

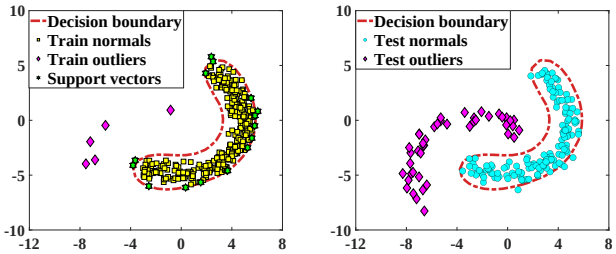
Author Name
Affiliation
email@example.com

Abstract

Anomaly detection methods based on support vector machine aim to establish an optimal decision boundary for discerning normal and abnormal samples. However, these methods often rely on the hinge loss function, resulting in a decision boundary determined by only a few samples (support vectors) near the boundary surface. This characteristic makes these methods susceptible to noise during the training phase, posing a challenge to their generalization performance. In this paper, we propose a novel **N**on-convex **A**nti-noise **M**odel (NAM) for anomaly detection, offering noise absorption and scaling properties to fortify the noise immunity of anomaly detection. Technically, NAM first introduces a new loss function named slant loss. Then, NAM formulates an objective function that incorporates regularization and loss terms to facilitate the learning of an optimal decision boundary. The introduction of the slant loss function makes NAM with the non-convexity, presenting a challenge in model solution. To address this problem, we further propose a non-convex decomposition solution method to obtain the optimal solution for NAM. The noise insensitivity of NAM is theoretically established by analyzing its parameters. Extensive experiments conducted on 25 real-world datasets and 10 synthetic datasets assess the noise immunity and effectiveness of NAM.

1 Introduction

Anomaly detection methods based on support vector machine (SVM) not only have a capability of handling complex non-linear relationships among samples by utilizing kernel functions, but also exhibit a solid mathematical foundation and provide support for generalization performance, thereby enhancing the reliability of these methods [Maziarka *et al.*, 2022]. Currently, SVM-based anomaly detection methods [Qiao *et al.*, 2023; Turkoz *et al.*, 2020] have found widespread application in various practical scenarios, such as digital healthcare [Raza *et al.*, 2023], fraud detection [Zhang *et al.*, 2022], and network security detection [Miao *et al.*, 2019].



(a) Training w/o noise (b) Testing w/o noise

(c) Training with noise (d) Testing with noise

Figure 1: Decision boundary based on hinge loss w/o and with noise.

SVM-based anomaly detection methods aim to establish an optimal decision boundary with a small number of training samples (support vectors), segregating all normal samples to one side of the boundary while placing a small number of outliers on the opposite side. To obtain this decision boundary, it is necessary to design an optimization objective that incorporates a loss function. Existing methods commonly employ the hinge loss function which imposes penalties on training samples (support vectors) that fall on the wrong side or are too close to the decision boundary. This property of the hinge loss function makes it sensitive to noise [Lee *et al.*, 2007]. Consider a synthetic dataset in Example 1.

Example 1. *When training samples do not contain noise (as shown in Figure 1(a)), the decision boundary can be well-determined by support vectors. During the testing phase, this decision boundary can accurately distinguish between anomalous and normal samples, as depicted in Figure 1(b). However, when training samples have noise, the hinge loss's focus on support vectors may lead to noisy samples being*

treated as support vectors. This can cause the decision boundary to adjust its position to accommodate these noisy samples, even if they are not representative of the overall data distribution, as illustrated in Figure 1(c). This adjustment results in suboptimal performance in distinguishing between outliers and normals during the testing phase (Figure 1(d)).

Currently, several anti-noise strategies have been introduced into SVM-based anomaly detection methods. The fundamental idea of these strategies is to assign a weight to each sample and uses the weighted samples to learn the optimal decision boundary. This requires additional runtime for data preprocessing to generate weights suitable for instances. Additionally, while there are differences in weight learning methods, they still use the hinge loss function in the objective function. This implies that the decision boundary of SVM-based anomaly detection methods is determined by a minority of samples on and near its surface, making the model still sensitive to noise.

Based on the above analysis, in this paper, we propose a novel **N**on-convex **A**nti-noise **M**odel (NAM). Specifically, NAM first develops a new slant loss function, which assigns additional penalties to correctly identified samples, allowing the optimal decision boundary to be determined by a larger number of samples collaboratively. The slant loss function introduces non-convexity to NAM, presenting a challenge in model solving. Therefore, we design a non-convex decomposition method to learn the optimal decision boundary of NAM. The scatter minimization and noise insensitivity of NAM are theoretically validated by analyzing the parameters in NAM.

The main contributions of this paper are as follows:

- We design a new loss function named slant loss, and then propose NAM, a con-convex anti-noise anomaly detection model based on this loss, to effectively alleviate the noise sensitivity of anomaly detection.
- We develop a non-convex decomposition method to learn the optimal decision boundary. The scatter minimization and noise insensitivity of NAM are also theoretically validated by analyzing NAM's parameters.
- We compare NAM with state-of-the-art anomaly detection methods on 25 real-world datasets and 10 synthetic datasets to validate the effectiveness and the noise immunity of NAM.

2 Related Work

Comprehensive reviews [Han *et al.*, 2022; Pang *et al.*, 2022; Ruff *et al.*, 2021] of the rich literature on anomaly detection is clearly beyond the capacity of this paper, so that we present some typical methods, and dedicated chapters in outlier analysis [Aggarwal, 2013] provide thorough treatments.

Supervised anomaly detection methods transform this task into the typical binary classification problem. Fully supervised anomaly detection is impractical due to the difficulty and cost of collecting labeled data. Currently researchers usually implement anomaly detection using existing classifiers. For example, ResNet [Gorishniy *et al.*, 2021] introduced a ResNet-like architecture for tabular based data. CatBoost [Prokhorenkova *et al.*, 2018] is a fast, scalable, high-performance gradient boosting on decision trees. Semi-supervised AD algorithms can capitalize the supervision from partial labels, while keeping the ability to detect unseen types of anomalies. DeepSAD [Ruff *et al.*, 2020] proposed a deep one-class method that improves DeepSVDD by penalizing the inverse of the distances of anomaly representation such that anomalies must be mapped further away from the hypersphere center. REPEN [Pang *et al.*, 2018] presented a neural network-based model that leverages transformed low-dimensional representation for random distance-based detectors. DevNet [Pang *et al.*, 2019] developed a neural network-based model uses a prior probability to enforce a statistical deviation score of input instances. FEAWAD [Zhou *et al.*, 2022] introduced a neural network-based model that incorporates the network architecture of DAGMM with the deviation loss of DevNet. GANomaly [Akcay *et al.*, 2018] introduced a GANbased method that defines the reconstruction error of the input instance as the anomaly score.

The two aforementioned methods exhibit excellent performance as they fully or partially rely on outliers' labels. However, in real scenarios, the challenge persists in leveraging domain expertise to define anomalies, posing an enduring difficulty for the task of anomaly detection. Therefore, several unsupervised anomaly detection methods have been proposed, categorizable into shallow and deep learning methods. These methods aim to detect outliers by proposing with different assumptions of data distribution. Shallow methods attempt to design models or to utilize statistical strategies to fit the data distribution. SVMs-based anomaly detection methods are one of the most typical shallow methods. Schölkopf *et al.* [Schölkopf *et al.*, 1999] and Tax *et al.* [Tax and Duin, 2004] proposed OCSVM and SVDD both of which aim to learn an optimal decision boundary that can segregate normal samples to one side. In addition, IForest [Liu *et al.*, 2008] developed an ensemble of trees to isolate the data points and defines the anomaly score as the distance of an individual instance to the root. ECOD [Li *et al.*, 2023] computed the empirical cumulative distribution per dimension of the input data, and then aggregates the tail probabilities per dimension for calculating the anomaly score. Deep learning methods design a neural network framework for outlier detection. DeepSVDD [Ruff *et al.*, 2018] trained a neural network to learn a transformation that minimizes the volume of a data-enclosing hypersphere in the output space, and calculates the anomaly score to the center of the hypersphere. DAGMM [Zong *et al.*, 2018] jointly optimized the deep autoencoder and the Gaussian mixture model in an end-to-end neural network fashion.

Note that the above methods all assume the purity of the data. However, in real-world scenarios, data often contain noise, and these methods fail to address this issue. Currently, Research on image and time series data is already emerging. For example, Zhou *et al.* [Zhou and Paffenroth, 2017] proposed RDA which can reconstructed the data and screend out the outliers and noise in the original data. Jiang *et al.* [Jiang *et al.*, 2022] introduced SoftPatch, which denoises

the data at the path level for image sensory anomaly detection. Liu *et al.* [Jiang *et al.*, 2022] designed SimpleNet for detecting and localizing anomalies. However, research on anomaly detection on numeric data in the presence of noise is limited. Therefore, this paper attempts to explore this issue with designing new loss function and optimization problem.

3 Proposed Model

3.1 Slant Loss Function

Taking into account the specific optimization objectives for anomaly detection tasks, we introduce a novel slant loss function, designed to be more advantageous for anomaly detection compared to the hinge loss. Given a training set $\mathbf{X} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n\}$, where each sample $\boldsymbol{x}_i \in \mathbb{R}^d$. The objective of SVM-based anomaly detection methods is to learn an optimal decision boundary from $\mathbf{X}$, such that it can divide the training samples into the same side as much as possible. Based on the above definitions, the general form of the slant loss function is expressed as:

$$\mathcal{R}_{\epsilon,\tau,s}(a) = \begin{cases} 0, & a < -\dfrac{\epsilon}{\tau} \\ \tau a + \epsilon, & -\dfrac{\epsilon}{\tau} \le a < 0 \\ a + \epsilon, & 0 \le a \le s \\ s + \epsilon, & a > s \end{cases} \tag{1}$$

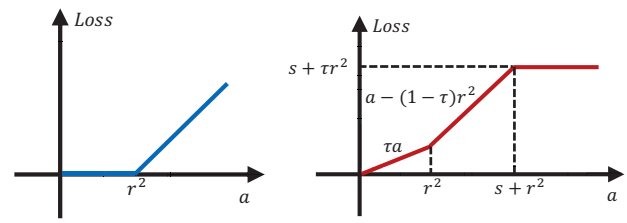
where a characterizes the functional margin of the sample. ϵ and s are constants, and $\tau \in [0, 1]$ denotes the slope of the linear function. To better illustrate the advantage of slant loss, we define the decision boundary as a hypersphere $h(\boldsymbol{c}, r)$, where $\boldsymbol{c}$ and r denote the centre and radius of the hypersphere, respectively. Then, the slant loss for hypersphere is represented as:

$$\mathcal{R}_{\tau,s}(a, r) = \begin{cases} \tau a, & a < r^2 \\ a - (1-\tau)r^2, & r^2 \le a \le s + r^2 \\ \tau r^2 + s, & a > s + r^2 \end{cases} \tag{2}$$

where $a = \|\phi(\boldsymbol{x}) - \boldsymbol{c}\|^2$, and $\phi(\cdot)$ is a mapping function.

Figure 2 visually illustrates the distinctions between the hinge loss and the proposed slant loss. Specifically, in the hinge loss, samples within the hypersphere ($a < r^2$) incur no penalties, while non-zero penalties are assigned to those situated on or outside the hypersphere ($a \ge r^2$). This implies that only samples positioned on or outside the hypersphere contribute to the decision boundary, potentially leading to bias of the hypersphere when these samples contain noise. In contrast, the slant loss handles samples differently. Those samples on the hypersphere's center ($a = 0$) face no penalties, while samples within the hypersphere receive increasing penalties τa with moving from the center to the hypersphere's surface. Simultaneously, samples outside the hypersphere incur higher penalties compared to those inside. Retaining the robust advantage for the noise outside of the hypersphere, training samples have constant penalties when $a > s + r^2$.

Different from hinge loss for hypersphere whose decision boundary is decided by only a few samples lying on and outside of the hypersphere, the slant loss involves samples within the hypersphere by assigning penalties and involving them in the hypersphere determination process. Consequently, in minimizing the cumulative penalty, the slant loss for hypersphere strives to encapsulate dense samples at the hypersphere's center while excluding sparse samples outside the hypersphere. This distinctive property enhances the robustness of the optimal hypersphere to noise.



(a) Hinge loss (b) Slant loss

Figure 2: Hinge loss and slant loss for hypersphere.

3.2 Design of NAM

With Equation (2), non-convex anti-noise model (NAM) is equivalent to the following form:

$$\min_{r,\boldsymbol{c}} \; r^2 + \frac{1}{\nu n} \sum_{i=1}^{n} \mathcal{R}_{\tau,s}(\boldsymbol{x}_i, \boldsymbol{c}, r) \tag{3}$$

where ν denotes the trade-off parameter. The $\mathcal{R}_{\tau,s}(\boldsymbol{x}_i, \boldsymbol{c}, r)$ can be broken down into the convex and concave functions, i.e., $\mathcal{R}_{\tau,s}(\boldsymbol{x}_i, \boldsymbol{c}, r) = \mathcal{H}_{\tau}(\boldsymbol{x}_i, \boldsymbol{c}, r) - \mathcal{H}_{\tau,s}(\boldsymbol{x}_i, \boldsymbol{c}, r)$, where

$$\mathcal{H}_{\tau}(\boldsymbol{x}_i, \boldsymbol{c}, r) = \begin{cases} \tau\|\phi(\boldsymbol{x}) - \boldsymbol{c}\|^2, & \|\phi(\boldsymbol{x}) - \boldsymbol{c}\|^2 < r^2 \\ \|\phi(\boldsymbol{x}) - \boldsymbol{c}\|^2 - (1-\tau)r^2, & \text{otherwise} \end{cases}$$

and

$$\mathcal{H}_{\tau,s}(\boldsymbol{x}_i, \boldsymbol{c}, r) = \begin{cases} 0, & \|\phi(\boldsymbol{x}) - \boldsymbol{c}\|^2 < s + r^2 \\ \|\phi(\boldsymbol{x}) - \boldsymbol{c}\|^2 - r^2 - s, & \text{otherwise} \end{cases}$$

Therefore, problem (3) can be reformulated as:

$$\min_{r,\boldsymbol{c}} \; r^2 + \frac{1}{\nu n} \sum_{i=1}^{n} \mathcal{H}_{\tau}(\boldsymbol{x}_i, \boldsymbol{c}, r) - \frac{1}{\nu n} \sum_{i=1}^{n} \mathcal{H}_{\tau,s}(\boldsymbol{x}_i, \boldsymbol{c}, r) \tag{4}$$

The above optimization problem (4) contains the convex part $f_{vex}(r, \boldsymbol{c}) = r^2 + 1/(\nu n) \sum_{i=1}^{n} \mathcal{H}_{\tau}(\boldsymbol{x}_i, \boldsymbol{c}, r)$ and concave part $f_{cov}(r, \boldsymbol{c}) = -1/(\nu n) \sum_{i=1}^{n} \mathcal{H}_{\tau,s}(\boldsymbol{x}_i, \boldsymbol{c}, r)$, which makes the optimization problem difficult to solve.

3.3 Optimization for NAM

The concave-convex procedure [Collobert *et al.*, 2006] is a powerful algorithm for solving non-convex optimization problems due to its simplicity in each iteration. In addition, this procedure has a lower computational cost compared to convex alternatives, resulting in greater benefits. The concave-convex procedure framework for the optimization problem (4) is constructed in Algorithm 1. We discuss the

Algorithm 1 Concave-convex procedure

1: Initialize $(r, \boldsymbol{c}, s)$, set $k := 0$
2: Construct and solve the optimization problem

$$\min_{r,\boldsymbol{c}} \quad f_{vex}(r, \boldsymbol{c}) + \langle f'_{cov}(r^{(k)}, \boldsymbol{c}^{(k)}), (r, \boldsymbol{c})\rangle \tag{5}$$

Get the solution $(r^{(k)*}, \boldsymbol{c}^{(k)*})$
3: If $(r^*, \boldsymbol{c}^*)$ satisfied convergence condition, the output solution will be $(r^*, \boldsymbol{c}^*) = (r^{(k)*}, \boldsymbol{c}^{(k)*})$, otherwise, set $k = k + 1$, go to step 2.

solution for problem (5) in Algorithm 1. $f'_{cov}(r, \boldsymbol{c})$ is a non-differentiable at the inflection point, so that concave-convex procedure remains valid when utilizing any upper-derivative of a concave function. Supposed $\boldsymbol{\varepsilon} = [\varepsilon_1, \cdots, \varepsilon_n]^\top$, where

$$\varepsilon_i = \begin{cases} -\dfrac{1}{\nu n}, & \|\phi(\boldsymbol{x}) - \boldsymbol{c}\|^2 > s + r^2 \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

Let ξ be a slack variable. Problem (4) is transformed as:

$$\begin{aligned} \min_{r,\boldsymbol{c}} \quad & r^2 + \frac{1}{\nu n}\sum_{i=1}^n \xi_i + \sum_{i=1}^n \varepsilon_i(\|\phi(\boldsymbol{x}_i) - \boldsymbol{c}\|^2 - r^2) \\ \text{s.t.} \quad & \|\phi(\boldsymbol{x}_i) - \boldsymbol{c}\|^2 - (1-\tau)r^2 \le \xi_i \\ & \|\phi(\boldsymbol{x}_i) - \boldsymbol{c}\|^2 \le \frac{1}{\tau}\xi_i, i = 1, \cdots, n \end{aligned}$$

To get the optimal variables r and $\boldsymbol{c}$, we first transform optimization problem into its dual form. Specifically, based on the Lagrangian multipliers $\alpha_i \ge 0$ and $\beta_i \ge 0$, Lagrangian function is:

$$\begin{aligned} \mathcal{L} = & r^2 + \frac{1}{\nu n}\sum_{i=1}^n \xi_i + \sum_{i=1}^n \varepsilon_i(\|\phi(\boldsymbol{x}_i) - \boldsymbol{c}\|^2 - r^2) \\ & + \sum_{i=1}^n \alpha_i(\|\phi(\boldsymbol{x}_i) - \boldsymbol{c}\|^2 - (1-\tau)r^2 - \xi_i) \\ & + \sum_{i=1}^n \beta_i(\|\phi(\boldsymbol{x}_i) - \boldsymbol{c}\|^2 - \frac{1}{\tau}\xi_i) \end{aligned} \tag{7}$$

The Karush-Kuhn-Tucker (KKT) conditions are defined as follows:

$$\begin{cases} \nabla_r \mathcal{L} = 1 - \sum_{i=1}^n(\varepsilon_i + (1-\tau)\alpha_i) = 0 \\ \nabla_{\xi_i}\mathcal{L} = \dfrac{1}{\nu n} - \alpha_i - \dfrac{1}{\tau}\beta_i = 0 \\ \nabla_{\boldsymbol{c}}\mathcal{L} = \boldsymbol{c} - \dfrac{\nu}{\tau}\sum_{i=1}^n(\varepsilon_i + (1-\tau)\alpha_i)\phi(\boldsymbol{x}_i) = 0 \\ \sum_{i=1}^n \alpha_i(\|\phi(\boldsymbol{x}_i) - \boldsymbol{c}\|^2 - (1-\tau)r^2 - \xi_i) = 0 \\ \sum_{i=1}^n \beta_i(\|\phi(\boldsymbol{x}_i) - \boldsymbol{c}\|^2 - \dfrac{1}{\tau}\xi_i) = 0 \end{cases} \tag{8}$$

Algorithm 2 NAM

Input: Training set $\mathbf{X} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n\}$, parameter ν
Output: $\boldsymbol{c}^*, r^*$

1: Select the proper value for s in anti-noise loss
2: Solve the problem (5)
(i) Initialize $\boldsymbol{\varepsilon}^{(1)}$ and set $k = 1$
(ii) Construct and solve the optimization problem (10) in the k-th iterative step. Compute $(r^{(k)}, \boldsymbol{c}^{(k)})$ based on KKT conditions. Compute $\boldsymbol{\varepsilon}^{(k)}$ based on Equation (6)
(iii) If $\boldsymbol{\varepsilon}^{(k)} = \boldsymbol{\varepsilon}^{(k-1)}$, stop this iteration. Get the solution $(r^*, \boldsymbol{c}^*) = (r^{(k)}, \boldsymbol{c}^{(k)})$

Let $\varepsilon_i + (1-\tau)\alpha_i = \gamma_i$. By putting Equation (8) into the Lagrangian function (7), we obtain the dual problem as:

$$\begin{aligned} \min_{\boldsymbol{\gamma},\boldsymbol{\varepsilon}} \quad & \sum_{i,j=1}^n (\gamma_i + \frac{\tau}{\nu n})\phi(\boldsymbol{x}_i)^\top\phi(\boldsymbol{x}_j)(\gamma_j + \frac{\tau}{\nu n}) \\ & - \sum_{i=1}^n(\gamma_i + \frac{\tau}{\nu n})\phi(\boldsymbol{x}_i)^\top\phi(\boldsymbol{x}_i) \\ \text{s.t.} \quad & \varepsilon_i \le \gamma_i \le \varepsilon_i + \frac{1-\tau}{\nu n} \\ & \sum_{i=1}^n \gamma_i = 1, i = 1, \cdots, n \end{aligned} \tag{9}$$

Let $\gamma_i + \dfrac{\tau}{\nu n} = \theta_i$. Utilizing Gaussian kernel function $\mathrm{K}(\cdot,\cdot)$, the optimization problem (9) will be rewritten as:

$$\begin{aligned} \min_{\boldsymbol{\gamma},\boldsymbol{\varepsilon}} \quad & \sum_{i,j=1}^n \theta_i \mathrm{K}(\boldsymbol{x}_i, \boldsymbol{x}_j)\theta_j - \sum_{i=1}^n \theta_i \mathrm{K}(\boldsymbol{x}_i, \boldsymbol{x}_i) \\ \text{s.t.} \quad & \varepsilon_i + \frac{\tau}{\nu n} \le \theta_i \le \varepsilon_i + \frac{1}{\nu n} \\ & \sum_{i=1}^n \theta_i = \frac{\tau}{\nu} + 1, i = 1, \cdots, n \end{aligned} \tag{10}$$

Based on the aforementioned formulas, NAM is constructed with the concave-convex procedure as shown in Algorithm 2. Finally, given a text sample $\boldsymbol{o}$, the decision function of NAM is obtained as:

$$\begin{aligned} g(\boldsymbol{o}) = & \text{sign}(\|\boldsymbol{o} - \boldsymbol{c}\|^2 - r^2) \\ = & \text{sign}(\mathrm{K}(\boldsymbol{o}, \boldsymbol{o}) - \frac{2\nu}{\tau}\sum_{i=1}^n(\theta_i - \frac{\tau}{\nu n})\mathrm{K}(\boldsymbol{o}, \boldsymbol{x}_i) \\ & + \frac{\nu^2}{\tau^2}\sum_{i,j=1}^n(\theta_i - \frac{\tau}{\nu n})\mathrm{K}(\boldsymbol{x}_i, \boldsymbol{x}_j)(\theta_j - \frac{\tau}{\nu n})) \end{aligned}$$

3.4 Scatter Minimization of NAM

NAM has a main property that is the scatter minimization in the centre of the hypersphere. Firstly, we divide the sample index into three subsets:

$$\begin{aligned} \mathbf{I}_{in} &= \{i : \|\phi(\boldsymbol{x}_i) - \boldsymbol{c}\|^2 - r^2 < 0\} \\ \mathbf{I}_{mid} &= \{i : T \ge \|\phi(\boldsymbol{x}_i) - \boldsymbol{c}\|^2 - r^2 \ge 0\} \\ \mathbf{I}_{out} &= \{i : \|\phi(\boldsymbol{x}_i) - \boldsymbol{c}\|^2 - r^2 > T\} \end{aligned}$$

Training samples in $\mathbf{I}_{in}$ lie inside the hypersphere, while that in $\mathbf{I}_{mid}$ locate outside the hypersphere. When the distance from the training sample to the centre is over the constant T, the training sample belong to the subset $\mathbf{I}_{out}$.

By utilizing the sum of square distance to a specific training sample, the scatter of $\boldsymbol{x}_i$ around the centre of the hypersphere is represented as $\sum_{i=1}^{n} \|\phi(\boldsymbol{x}_i) - \boldsymbol{c}\|^2$. Then, we can get the objective function:

$$\min_{r,\boldsymbol{c}} \ r^2 + \frac{1}{\nu_1 n} \sum_{i=1}^{n} \|\phi(\boldsymbol{x}_i) - \boldsymbol{c}\|^2 \tag{11}$$

Observing the problem (11), the first term aims to minimize the radius r of the hypersphere, which makes the optimal hypersphere compact. The second term denotes minimizing the scatter around the centre of the hypersphere, so that this term can be extended to the slant loss function $\mathcal{R}_{\tau,s}(\boldsymbol{x}_i, \boldsymbol{c}, r)$. Specifically, given the function $\frac{1}{\nu_2 n} \max\{0, \|\phi(\boldsymbol{x}) - \boldsymbol{c}\|^2 - r^2, s\}$, we have:

$$\begin{aligned}
&\min_{r,\boldsymbol{c}} \quad r^2 + \frac{1}{\nu_1 n} \sum_{i=1}^{n} \|\phi(\boldsymbol{x}_i) - \boldsymbol{c}\|^2 \\
&\qquad + \frac{1}{\nu_2 n} \max\{0, \|\phi(\boldsymbol{x}) - \boldsymbol{c}\|^2 - r^2, s\} \\
\Rightarrow &\min_{r,\boldsymbol{c}} \quad r^2 + (\frac{1}{\nu_1 n} + \frac{1}{\nu_2 n})(\frac{\nu_2}{\nu_1 + \nu_2} \sum_{i \in \mathbf{I}_{in}} \|\phi(\boldsymbol{x}_i) - \boldsymbol{c}\|^2 \\
&\qquad + \sum_{i \in \mathbf{I}_{mid}} (\|\phi(\boldsymbol{x}_i) - \boldsymbol{c}\|^2 - \frac{\nu_1}{\nu_1 + \nu_2} r^2) \\
&\qquad + \sum_{i \in \mathbf{I}_{out}} (s + r^2 - \frac{\nu_1}{\nu_1 + \nu_2} r^2))
\end{aligned} \tag{12}$$

Let $\frac{1}{\nu_1 n} + \frac{1}{\nu_2 n} = \frac{1}{\nu n}$ and $1 - \tau = \frac{\nu_1}{\nu_1 + \nu_2}$. Problem (12) is transformed as:

$$\begin{aligned}
\min_{r,\boldsymbol{c}} \ r^2 &+ \frac{1}{\nu n}(\sum_{i \in \mathbf{I}_{in}} \tau \|\phi(\boldsymbol{x}_i) - \boldsymbol{c}\|^2 + \sum_{i \in \mathbf{I}_{out}} (s + \tau r^2) \\
&+ \sum_{i \in \mathbf{I}_{mid}} (\|\phi(\boldsymbol{x}_i) - \boldsymbol{c}\|^2 - (1 - \tau) r^2)
\end{aligned}$$

From the above problem (12), we can observe that NAM takes three problems into account, including the scatter minimization, the miss detecting error minimization, and the outlier penalty bias. These three problems can help to achieve a better robustness.

3.5 Noise Insensitivity of NAM

We introduce following two propositions to demonstrate the noise insensitivity of NAM.

Proposition 1. *For the optimization problem (10), the result should satisfy* $n(\varepsilon_i + \frac{1}{\nu n}) \geq \frac{\tau}{\nu} + 1$.

Proof. From Equation (10), we have $\frac{\tau}{\nu} + 1 = \sum_{i \in \mathbf{I}_{in}} \theta_i + \sum_{i \in \mathbf{I}_{mid}} \theta_i + \sum_{i \in \mathbf{I}_{out}} \theta_i$. Let $\boldsymbol{x}_q$ denote an arbitrary outlier in $\mathbf{I}_{mid} \cup \mathbf{I}_{out}$. Define the number of the training sample in $\mathbf{I}_{mid}$ and $\mathbf{I}_{out}$ as m. Based on Equation (8), we have $\beta_i = 0$ and $\theta_i = \varepsilon_i + \frac{1}{\nu n}$. Therefore, we have $\frac{\tau}{\nu} + 1 - \sum_{i \in \mathbf{I}_{in}} \theta_i = m(\varepsilon_i + \frac{1}{\nu n})$. Then, we have $m(\varepsilon_i + \frac{1}{\nu n}) \geq \frac{\tau}{\nu} + 1 - (\varepsilon_i + \frac{1}{\nu n})(n - m)$. Finally, we can get $n(\varepsilon_i + \frac{1}{\nu n}) \geq \frac{\tau}{\nu} + 1$. □

Proposition 2. *Define the number of the training sample in* $\mathbf{I}_{mid}$ *and* $\mathbf{I}_{out}$ *as* m. *We have* $m \leq \frac{n(\tau + \nu)}{\varepsilon_i \nu n + 1}$.

Proof. Let $\boldsymbol{x}_q$ denote an arbitrary outlier in $\mathbf{I}_{mid} \cup \mathbf{I}_{out}$. Based on Equation (8), we have $\beta_i = 0$ and $\theta_i = \varepsilon_i + \frac{1}{\nu n}$. Therefore, we have $\frac{\tau}{\nu} + 1 - \sum_{i \in \mathbf{I}_{in}} \theta_i = m(\varepsilon_i + \frac{1}{\nu n})$. Then, we have $\frac{\tau}{\nu} + 1 \geq m(\varepsilon_i + \frac{1}{\nu n})$. Finally, we can get $m \leq \frac{n(\tau + \nu)}{\varepsilon_i \nu n + 1}$. □

4 Experiments and Results

All experiments are conducted on a PC with AMD 3990X 64-Core Processor 2.90 GHz CPU, 256 GB memory, running Windows10 Professional. NAM is implemented in MATLAB (R2020a).

4.1 Experimental Setting

Datasets. We evaluate all methods on 25 real-world datasets[1] and 10 synthetic datasets (banana and circle datasets with noise ratios from $\{0.1, 0.3, 0.5, 0.7, 0.9\}$). The statistics of datasets are summarized in Table 1.

Baselines. NAM is compared with 18 state-of-the-art anomaly detection methods, including I-Forst [Liu *et al.*, 2008], OCSVM [Schölkopf *et al.*, 1999], CBLOF [He *et al.*, 2003], COF [Tang *et al.*, 2002], COPOD [Li *et al.*, 2020], ECOD [Li *et al.*, 2023], PinSVD-D [Zhong *et al.*, 2022], HBOS [Goldstein and Dengel, 2012], KNN [Ramaswamy *et al.*, 2000], LODA [Pevný, 2016], LOF [Breunig *et al.*, 2000], SVDD [Tax and Duin, 2004], PCA [Shyu *et al.*, 2003], DeepSVDD [Ruff *et al.*, 2018], SOD [Kriegel *et al.*, 2009], DAGMM [Zong *et al.*, 2018], NS-NN [Sipple, 2020], Q-DEMDE [Useche *et al.*, 2022].

Metric. We conventionally use the Area Under Receiver Operating Characteristic Curve (AUC) as the performance metric.

4.2 Experimental Results

Performance on Real-world Datasets. Table 4 presents the AUC values for various anomaly detection methods. Experimental results indicate that across 25 real-world datasets, NAM achieves the optimal performance in 22 and the suboptimal performance in 3, particularly attaining a 100% AUC on datasets such as "lympho", "wine", "wdbc", "musk", and

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

Table 1: Statistics of 25 real-world datasets and 10 synthetics datasets.

Dataset	# Samples	# Attributes	# Outliers	% Anomaly	Category
Hepatitis	80	19	13	16.25	Healthcare
Lympho	148	18	6	4.05	Healthcare
wine	129	13	10	7.75	Chemistry
vertebral	240	6	30	12.50	Biology
breastw	683	9	239	34.99	Healthcare
glass	214	7	9	4.21	Forensic
pima	768	8	268	34.9	Healthcare
yeast	1484	8	507	34.16	Biology
stamps	340	9	31	9.12	Document
wpbc	198	33	47	23.74	Healthcare
letter	1600	32	100	6.25	Image
cardio	2114	21	466	22.04	Healthcare
waveform	3443	21	100	2.90	Physics
wdbc	367	30	10	2.72	Healthcare
fault	1941	27	673	34.67	Physical
vowels	1456	12	50	3.43	Linguistics
wilt	4819	5	257	5.33	Botany
ionosphere	351	33	126	35.9	Oryctognosy
spambase	4207	57	1679	39.91	Document
satimage2	5803	36	71	1.22	Astronautics
internetAds	1966	1555	368	18.72	Image
musk	3062	166	97	3.17	Chemistry
satellite	6435	36	2036	31.64	Astronautics
landsat	6435	36	1333	20.71	Astronautics
pendigits	6870	16	156	2.27	Image
Dataset	# Samples	# Attributes	# Outliers	% Anomaly	Category
banana-0.1	2000	2	30	1.5	Synthetics
banana-0.3	2000	2	30	1.5	Synthetics
banana-0.5	2000	2	30	1.5	Synthetics
banana-0.7	2000	2	30	1.5	Synthetics
banana-0.9	2000	2	30	1.5	Synthetics
circle-0.1	2000	2	30	1.5	Synthetics
circle-0.3	2000	2	30	1.5	Synthetics
circle-0.5	2000	2	30	1.5	Synthetics
circle-0.7	2000	2	30	1.5	Synthetics
circle-0.9	2000	2	30	1.5	Synthetics

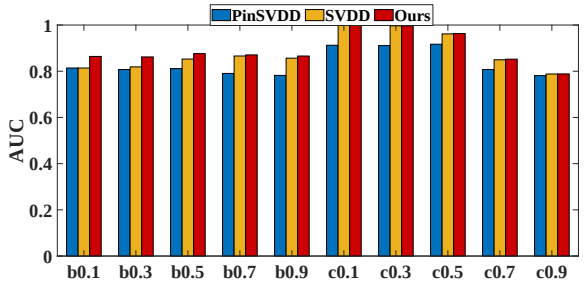


Figure 5: AUC on 10 synthetic datasets ($\tau, s, \nu = 0.1$).

"pendigits". This underscores NAM's ability to identify outliers across all thresholds. The robustness of NAM, empowered by the employed slant loss function, is evident in its effective handling of noise interference present in real-world datasets. Given the necessity for anomaly detection to address specific data patterns, NAM accurately captures the normal behavior of normal samples, adapting better to the distribution and characteristics of real-world data. However, on datasets such as "vertebral", "wpbc", and "annthyroid", most methods achieve AUC results around 60%. This may be attributed to insufficient features in the dataset to distinguish between normal and anomalous samples, leading to the misclassification of outliers as normal ones in cases of sample imbalance. Furthermore, compared to other baselines, NAM exhibits smaller standard deviations, indicating a more concentrated distribution of experimental results with minimal differences between data points. This highlights the model's high stability across various experimental runs.

Performance on Synthetic Datasets. We conduct a com-

	hepatitis	lympho	wine	vertebral	breastw	glass	pima	yeast
IForest	0.698±0.011	0.998±0.003	0.804±0.051	0.367±0.038	0.983±0.003	0.771±0.017	0.729±0.038	0.378±0.023
OCSVM	0.677±0.024	0.995±0.006	0.731±0.045	0.380±0.032	0.803±0.044	0.354±0.104	0.669±0.039	0.410±0.032
CBLOF	0.664±0.053	0.998±0.001	0.259±0.056	0.414±0.041	0.968±0.003	**0.829±0.051**	0.714±0.025	0.448±0.009
COF	0.415±0.011	0.908±0.037	0.444±0.051	0.487±0.037	0.388±0.012	0.722±0.079	0.610±0.016	0.445±0.020
COPOD	0.821±0.012	0.995±0.004	0.887±0.019	0.256±0.033	**0.997±0.001**	0.724±0.039	0.691±0.022	0.370±0.022
ECOD	0.752±0.025	0.995±0.003	0.772±0.026	0.407±0.034	0.992±0.002	0.658±0.045	0.631±0.022	0.438±0.024
PinSVDD	0.690±0.030	0.982±0.025	**1.000±0.000**	0.450±0.107	0.986±0.017	0.351±0.035	0.678±0.062	**0.523±0.037**
HBOS	0.799±0.013	0.995±0.005	0.914±0.003	0.286±0.027	0.989±0.003	0.772±0.014	0.711±0.020	0.396±0.022
KNN	0.528±0.062	0.559±0.148	0.450±0.047	0.338±0.073	0.970±0.005	0.813±0.081	0.734±0.017	0.391±0.024
LODA	0.649±0.134	0.855±0.131	0.901±0.042	0.306±0.063	0.985±0.007	0.731±0.065	0.659±0.033	0.446±0.011
LOF	0.380±0.016	0.899±0.016	0.377±0.055	0.493±0.058	0.402±0.049	0.692±0.050	0.657±0.015	0.453±0.009
SVDD	0.613±0.129	**1.000±0.000**	**1.000±0.000**	0.454±0.148	0.984±0.012	0.391±0.360	0.682±0.023	0.451±0.021
PCA	0.759±0.023	0.998±0.002	0.844±0.013	0.371±0.026	0.951±0.004	0.663±0.041	0.708±0.044	0.412±0.031
SOD	0.687±0.042	0.721±0.160	0.451±0.044	0.402±0.056	0.943±0.007	0.732±0.077	0.612±0.032	0.424±0.032
DeepSVDD	0.574±0.176	0.719±0.072	0.643±0.135	0.378±0.038	0.634±0.052	0.557±0.111	0.495±0.033	0.434±0.070
DAGMM	0.548±0.059	0.721±0.211	0.617±0.209	0.532±0.053	-	0.761±0.120	0.559±0.017	0.411±0.017
NS-NN	-	-	-	0.329±0.076	0.979±0.007	-	0.686±0.035	0.400±0.035
NS-RF	-	-	-	0.286±0.095	0.976±0.014	-	0.643±0.037	0.421±0.031
Q-DEMDE	0.587±0.013	0.450±0.006	0.429±0.000	0.482±0.002	0.438±0.001	0.562±0.024	0.475±0.001	0.440±0.001
NAM	**0.867±0.183**	**1.000±0.000**	**1.000±0.000**	**0.546±0.135**	0.992±0.002	0.817±0.100	**0.742±0.058**	0.472±0.041

Table 2: AUC of 20 anomaly detection methods on 8 real-world datasets ($\tau, s, \nu = 0.1$, bold and underline indicate the highest and second highest AUCs, respectively).

	stamps	wpbc	letter	cardio	wavform	wdbc	fault	vowels
IForest	0.912±0.018	0.466±0.046	0.611±0.018	0.676±0.046	0.715±0.026	0.989±0.006	0.570±0.032	0.739±0.042
OCSVM	0.839±0.052	0.454±0.029	0.462±0.007	0.779±0.015	0.563±0.025	0.989±0.002	0.477±0.040	0.616±0.022
CBLOF	0.682±0.072	0.448±0.082	0.756±0.035	0.645±0.010	0.724±0.067	0.993±0.005	0.641±0.020	0.899±0.000
COF	0.538±0.076	0.459±0.032	0.800±0.011	0.538±0.018	0.726±0.016	0.963±0.024	0.621±0.032	0.940±0.024
COPOD	0.934±0.014	0.493±0.043	0.543±0.011	0.670±0.020	0.750±0.019	0.994±0.003	0.439±0.030	0.532±0.029
ECOD	0.879±0.020	0.466±0.044	0.560±0.016	0.784±0.017	0.624±0.025	0.972±0.005	0.454±0.021	0.622±0.005
PinSVDD	0.827±0.069	0.443±0.141	0.871±0.017	0.836±0.014	0.691±0.103	**1.000±0.000**	0.602±0.061	0.827±0.039
HBOS	0.907±0.017	0.512±0.047	0.597±0.022	0.609±0.019	0.688±0.024	0.995±0.001	0.513±0.064	0.722±0.012
KNN	0.686±0.056	0.466±0.078	0.862±0.020	0.562±0.026	0.738±0.046	0.917±0.079	0.730±0.022	0.973±0.007
LODA	0.872±0.044	0.493±0.033	0.502±0.048	0.737±0.051	0.601±0.063	0.983±0.002	0.417±0.033	0.704±0.027
LOF	0.513±0.121	0.414±0.074	0.845±0.008	0.595±0.028	0.733±0.060	0.890±0.040	0.589±0.038	0.931±0.015
SVDD	0.568±0.117	0.454±0.193	0.841±0.010	0.844±0.013	0.800±0.054	**1.000±0.000**	0.595±0.024	0.791±0.061
PCA	0.915±0.019	0.460±0.036	0.503±0.019	0.747±0.017	0.655±0.021	0.991±0.003	0.460±0.033	0.653±0.022
SOD	0.735±0.036	0.512±0.063	0.841±0.008	0.517±0.026	0.686±0.046	0.919±0.036	0.681±0.028	0.927±0.019
DeepSVDD	0.595±0.036	0.440±0.048	0.559±0.064	0.550±0.179	0.543±0.018	0.686±0.173	0.476±0.067	0.499±0.066
DAGMM	0.889±0.047	0.478±0.073	0.504±0.054	0.620±0.017	0.493±0.052	0.767±0.136	0.459±0.055	0.606±0.012
NS-NN	-	0.520±0.115	0.655±0.054	0.611±0.049	0.767±0.014	-	0.530±0.016	0.941±0.028
NS-RF	-	**0.615±0.096**	0.632±0.033	0.586±0.037	0.622±0.029	-	0.395±0.023	0.944±0.036
Q-DEMDE	0.616±0.015	0.422±0.002	0.529±0.000	0.493±0.001	0.456±0.001	0.804±0.000	0.584±0.013	0.749±0.001
NAM	**0.949±0.028**	0.532±0.078	**0.942±0.019**	**0.898±0.022**	**0.806±0.040**	**1.000±0.000**	**0.856±0.021**	**0.989±0.017**

Table 3: AUC of 20 anomaly detection methods on 8 real-world datasets ($\tau, s, \nu = 0.1$, bold and underline indicate the highest and second highest AUCs, respectively).

parative analysis on synthetic datasets with varying levels of noise to illustrate the robustness of NAM in comparison to SVDD and PinSVDD. As depicted in Figure 5, NAM consistently outperforms SVDD and PinSVDD in terms of AUC on all synthetic datasets. Notably, on datasets such as "banana0.1" and "banana0.5", NAM exhibits a significant 5% improvement in AUC. We observe that even as the noise rate increases on banana datasets, NAM maintains a relatively stable AUC, validating its robustness to noise. On circle datasets, both NAM and SVDD experience a decline in AUC with increasing noise rates. To investigate this phenomenon, we provide visualizations of SVDD and NAM on banana and circle datasets shown in Figures 3 and 4. The visualization results for the remaining datasets are listed in the Appendix. It is evident that, compared to SVDD, NAM incorporates more samples in determining the hyperplane, and the hypersphere volume of NAM is smaller than that of SVDD. This highlights the advantage of the slant loss function, ensuring that NAM learns a hypersphere with the minimum volume while keeping all samples as close to the center of the hypersphere as possible. Consequently, during the testing phase, NAM can more accurately distinguish between normal and anomalous samples compared to SVDD. However, in scenarios with a high noise rate, anomalous samples become mixed with normal samples, making it challenging for the hypersphere to differentiate them. Consequently, the AUC of NAM, PinSVD-

	wilt	iono	spambase	satimage2	internetAds	musk	satellite	landsat	pendigits
IForest	0.419±0.037	0.845±0.016	0.648±0.012	0.992±0.007	0.690±0.012	**1.000±0.000**	0.704±0.012	0.476±0.032	0.948±0.010
OCSVM	0.313±0.017	0.759±0.021	0.525±0.010	0.973±0.011	0.683±0.026	0.806±0.094	0.590±0.014	0.361±0.023	0.938±0.001
CBLOF	0.325±0.027	0.907±0.009	0.550±0.010	0.998±0.001	0.706±0.025	**1.000±0.000**	0.713±0.013	0.635±0.004	0.904±0.045
COF	0.497±0.005	0.868±0.030	0.410±0.002	0.567±0.092	0.638±0.071	0.387±0.012	0.547±0.007	0.535±0.005	0.451±0.041
COPOD	0.334±0.013	0.793±0.025	0.701±0.004	0.972±0.010	0.671±0.029	0.942±0.008	0.632±0.009	0.415±0.022	0.907±0.007
ECOD	0.363±0.026	0.732±0.018	0.669±0.004	0.963±0.009	0.671±0.029	0.954±0.004	0.581±0.010	0.361±0.023	0.930±0.005
PinSVDD	0.166±0.015	0.821±0.060	0.339±0.011	0.242±0.025	0.746±0.022	**1.000±0.000**	0.656±0.021	0.544±0.026	0.925±0.021
HBOS	0.325±0.031	0.625±0.030	0.647±0.013	0.976±0.014	0.680±0.027	**1.000±0.000**	0.748±0.003	0.551±0.018	0.930±0.010
KNN	0.484±0.019	0.883±0.042	0.534±0.007	0.926±0.009	0.700±0.017	0.699±0.028	0.652 ±0.004	0.580±0.003	0.729±0.005
LODA	0.264±0.020	0.784±0.054	0.420±0.127	0.976±0.013	0.554±0.055	0.951±0.023	0.620±0.018	0.382±0.016	0.891±0.015
LOF	0.507±0.021	0.906±0.021	0.433±0.008	0.474±0.063	0.658±0.048	0.412±0.084	0.559±0.012	0.539±0.007	0.480±0.029
SVDD	0.832±0.019	0.812±0.087	0.782±0.019	**1.000±0.000**	0.709±0.017	**1.000±0.000**	0.906±0.009	0.392±0.025	0.888±0.032
PCA	0.204±0.021	0.792±0.013	0.547±0.003	0.976±0.012	0.617±0.039	**1.000±0.000**	0.596±0.014	0.358±0.023	0.937±0.004
SOD	0.533±0.012	0.865±0.041	0.523±0.014	0.831±0.018	0.619±0.020	0.741±0.060	0.640±0.001	0.595±0.012	0.663±0.032
DeepSVDD	0.459±0.021	0.539±0.034	0.457±0.072	0.548±0.112	0.626±0.025	0.384±0.070	0.570±0.033	0.627±0.019	0.249±0.071
DAGMM	0.373±0.057	0.734±0.039	-	0.963±0.016	-	0.768±0.228	0.623±0.035	0.439±0.025	0.642±0.190
NS-NN	0.348±0.028	**0.966±0.014**	0.564±0.018	0.999±0.001	-	0.625±0.060	0.699±0.025	0.509±0.029	0.881±0.015
NS-RF	0.370±0.017	0.915±0.029	0.395±0.039	0.982±0.025	-	0.305±0.070	0.646±0.019	0.403±0.017	0.935±0.012
Q-DEMDE	0.513±0.001	0.542±0.002	0.482±0.000	0.492±0.002	0.579±0.001	0.467±0.000	0.584±0.000	0.560±0.001	0.527±0.001
NAM	**0.878±0.028**	0.965±0.020	**0.908±0.044**	**1.000±0.000**	**0.886±0.021**	**1.000±0.000**	**0.999±0.002**	**0.791±0.019**	**1.000±0.000**

Table 4: AUC of 20 anomaly detection methods on 9 real-world datasets ($\tau, s, \nu = 0.1$, bold and underline indicate the highest and second highest AUCs, respectively).

D, and SVDD experiences a decline.

Parameter Sensitivity Analysis. The parameter τ controls the penalty for samples between the center and the surface of the hypersphere, while s determines the bandwidth of constant penalties for samples outside the hypersphere. As illustrated in Figure 6, with τ fixed, increasing the bandwidth results in an initial increase followed by a decrease in AUC. This indicates that an appropriate bandwidth is crucial for NAM to achieve high performance, as either too narrow or too wide can adversely affect the model's performance. Conversely, when s is fixed and τ is increased, it is observed that NAM's performance gradually decreases with the increasing slope. Performance becomes extremely low when the slope reaches 0.9. This suggests that samples within the hypersphere should only serve as auxiliary decisions. The penalty for these samples should not exceed that of samples on the hypersphere's surface and outside the hypersphere.

5 Conclusion

In this paper, we introduce a novel anomaly detection model named NAM, characterized by its effectiveness and robustness, enhancing the precision of anomaly detection. NAM addresses the specific task of anomaly detection by devising a new slant loss function, facilitating the broader consideration of samples when learning the optimal decision boundary. This characteristic imbues the method with robustness but introduces non-convexity. Consequently, we propose a non-convex decomposition solution method to attain the optimal solution for the method. Extensive experiments assess the robustness and effectiveness of NAM. In future work, we aim to delve deeper into achieving accurate anomaly detection in the presence of noise in both attributes and labels.

References

[Aggarwal, 2013] Charu C. Aggarwal. *Outlier Analysis*. Springer, 2013.

[Akcay *et al.*, 2018] Samet Akcay, Amir Atapour Abarghouei, and Toby P. Breckon. Ganomaly: Semi-supervised anomaly detection via adversarial training. In *Proceedings of the 2018 Asian Conference on Computer Vision Computer Vision, ACCV*, pages 622–637, 2018.

[Breunig *et al.*, 2000] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. Lof: Identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, SIGMOD*, page 93–104, 2000.

[Collobert *et al.*, 2006] Ronan Collobert, Fabian H. Sinz, Jason Weston, and Léon Bottou. Trading convexity for scalability. In *Proceedings of the 2006 International Conference on Machine Learning, ICML*, pages 201–208, 2006.

[Goldstein and Dengel, 2012] Markus Goldstein and Andreas Dengel. Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm. 2012.

[Gorishniy *et al.*, 2021] Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Revisiting deep learning models for tabular data. In *Proceedings of the 2021 Annual Conference on Neural Information Processing Systems, NIPS*, pages 18932–18943, 2021.

[Han *et al.*, 2022] Songqiao Han, Xiyang Hu, Hailiang Huang, Minqi Jiang, and Yue Zhao. Adbench: Anomaly detection benchmark. In *Proceedings of the 2022 Annual Conference on Neural Information Processing Systems, NIPS*, 2022.

[He *et al.*, 2003] Zengyou He, Xiaofei Xu, and Shengchun Deng. Discovering cluster-based local outliers. *Pattern Recognit. Lett.*, 24(9-10):1641–1650, 2003.

[Jiang *et al.*, 2022] Xi Jiang, Jianlin Liu, Jinbao Wang, Qiang Nie, Kai Wu, Yong Liu, Chengjie Wang, and Feng Zheng. Softpatch: Unsupervised anomaly detection with noisy data. In *Proceedings of the 2022 Annual Conference on Neural Information Processing Systems, NIPS*, 2022.

[Kriegel *et al.*, 2009] Hans-Peter Kriegel, Peer Kröger, Erich Schubert, and Arthur Zimek. Outlier detection in axis-parallel subspaces of high dimensional data. In *Proceedings of the 2009 Pacific-Asia Conference on Knowledge Discovery and Data Mining, PAKDD*, pages 831–838, 2009.

[Lee *et al.*, 2007] Ki Young Lee, Dae-Won Kim, Kwang Hyung Lee, and Doheon Lee. Density-induced support vector data description. *IEEE Trans. Neural Networks*, 18(1):284–289, 2007.

[Li *et al.*, 2020] Zheng Li, Yue Zhao, Nicola Botta, Cezar Ionescu, and Xiyang Hu. COPOD: copula-based outlier detection. In *Proceedings of the 2020 International Conference on Data Mining, ICDM*, pages 1118–1123, 2020.

[Li *et al.*, 2023] Zheng Li, Yue Zhao, Xiyang Hu, Nicola Botta, Cezar Ionescu, and George H. Chen. ECOD: unsupervised outlier detection using empirical cumulative distribution functions. *IEEE Trans. Knowl. Data Eng.*, 35(12):12181–12193, 2023.

[Liu *et al.*, 2008] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *Proceedings of the 2008 International Conference on Data Mining, ICDM*, pages 413–422, 2008.

[Maziarka *et al.*, 2022] Lukasz Maziarka, Marek Smieja, Marcin Sendera, Lukasz Struski, Jacek Tabor, and Przemyslaw Spurek. Oneflow: One-class flow for anomaly detection based on a minimal volume region. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(11):8508–8519, 2022.

[Miao *et al.*, 2019] Xuedan Miao, Ying Liu, Haiquan Zhao, and Chunguang Li. Distributed online one-class support vector machine for anomaly detection over networks. *IEEE Trans. Cybern.*, 49(4):1475–1488, 2019.

[Pang *et al.*, 2018] Guansong Pang, Longbing Cao, Ling Chen, and Huan Liu. Learning representations of ultrahigh-dimensional data for random distance-based outlier detection. In *Proceedings of the 2018 International Conference on Knowledge Discovery and Data Mining, SIGKDD*, pages 2041–2050, 2018.

[Pang *et al.*, 2019] Guansong Pang, Chunhua Shen, and Anton van den Hengel. Deep anomaly detection with deviation networks. In *Proceedings of the 2019 International Conference on Knowledge Discovery and Data Mining, SIGKDD*, pages 353–362, 2019.

[Pang *et al.*, 2022] Guansong Pang, Chunhua Shen, Longbing Cao, and Anton van den Hengel. Deep learning for anomaly detection: A review. *ACM Comput. Surv.*, 54(2):38:1–38:38, 2022.

[Pevný, 2016] Tomás Pevný. Loda: Lightweight on-line detector of anomalies. *Mach. Learn.*, 102(2):275–304, 2016.

[Prokhorenkova *et al.*, 2018] Liudmila Ostroumova Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. Catboost: unbiased boosting with categorical features. In *Proceedings of the 2018 Annual Conference on Neural Information Processing Systems, NIPS*, pages 6639–6649, 2018.

[Qiao *et al.*, 2023] Yan Qiao, Kui Wu, and Peng Jin. Efficient anomaly detection for high-dimensional sensing data with one-class support vector machine. *IEEE Trans. Knowl. Data Eng.*, 35(1):404–417, 2023.

[Ramaswamy *et al.*, 2000] Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim. Efficient algorithms for mining outliers from large data sets. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, SIGMOD*, pages 427–438, 2000.

[Raza *et al.*, 2023] Ali Raza, Kim Phuc Tran, Ludovic Koehl, and Shujun Li. Anofed: Adaptive anomaly detection for digital health using transformer-based federated learning and support vector data description. *Eng. Appl. Artif. Intell.*, 121:106051, 2023.

[Ruff *et al.*, 2018] Lukas Ruff, Nico Görnitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Robert A. Vandermeulen, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In *Proceedings of the 2018 International Conference on Machine Learning, ICML*, pages 4390–4399, 2018.

[Ruff *et al.*, 2020] Lukas Ruff, Robert A. Vandermeulen, Nico Görnitz, Alexander Binder, Emmanuel Müller, Klaus-Robert Müller, and Marius Kloft. Deep semi-supervised anomaly detection. In *Proceedings of the 2020 International Conference on Learning Representations, ICLR*, 2020.

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

[Schölkopf *et al.*, 1999] Bernhard Schölkopf, Robert C. Williamson, Alexander J. Smola, John Shawe-Taylor, and John C. Platt. Support vector method for novelty detection. In *Proceedings of the 1999 Annual Conference on Neural Information Processing Systems, NIPS*, pages 582–588, 1999.

[Shyu *et al.*, 2003] Mei-Ling Shyu, Shu-Ching Chen, Kanoksri Sarinnapakorn, and LiWu Chang. A novel anomaly detection scheme based on principal component classifier. *Proceedings of the 2003 International Conference on Data Mining, ICDM*, 2003.

[Sipple, 2020] John Sipple. Interpretable, multidimensional, multimodal anomaly detection with negative sampling for detection of device failure. In *Proceedings of the 2020 International Conference on Machine Learning, ICML*, pages 9016–9025, 2020.

[Tang *et al.*, 2002] Jian Tang, Zhixiang Chen, Ada Wai-chee Fu, and David W. Cheung. *Enhancing Effectiveness of Outlier Detections for Low Density Patterns*, page 535–548. 2002.

[Tax and Duin, 2004] David M. J. Tax and Robert P. W. Duin. Support vector data description. *Mach. Learn.*, 54(1):45–66, 2004.

[Turkoz *et al.*, 2020] Mehmet Turkoz, Sangahn Kim, Youngdoo Son, Myong K. Jeong, and Elsayed A. Elsayed. Generalized support vector data description for anomaly detection. *Pattern Recognit.*, 100:107119, 2020.

[Useche *et al.*, 2022] Diego H. Useche, Oscar A. Bustos-Brinez, Joseph A. Gallego-Mejia, and Fabio A. Gonz'alez. Quantum density estimation with density matrices: Application to quantum anomaly detection. 2022.

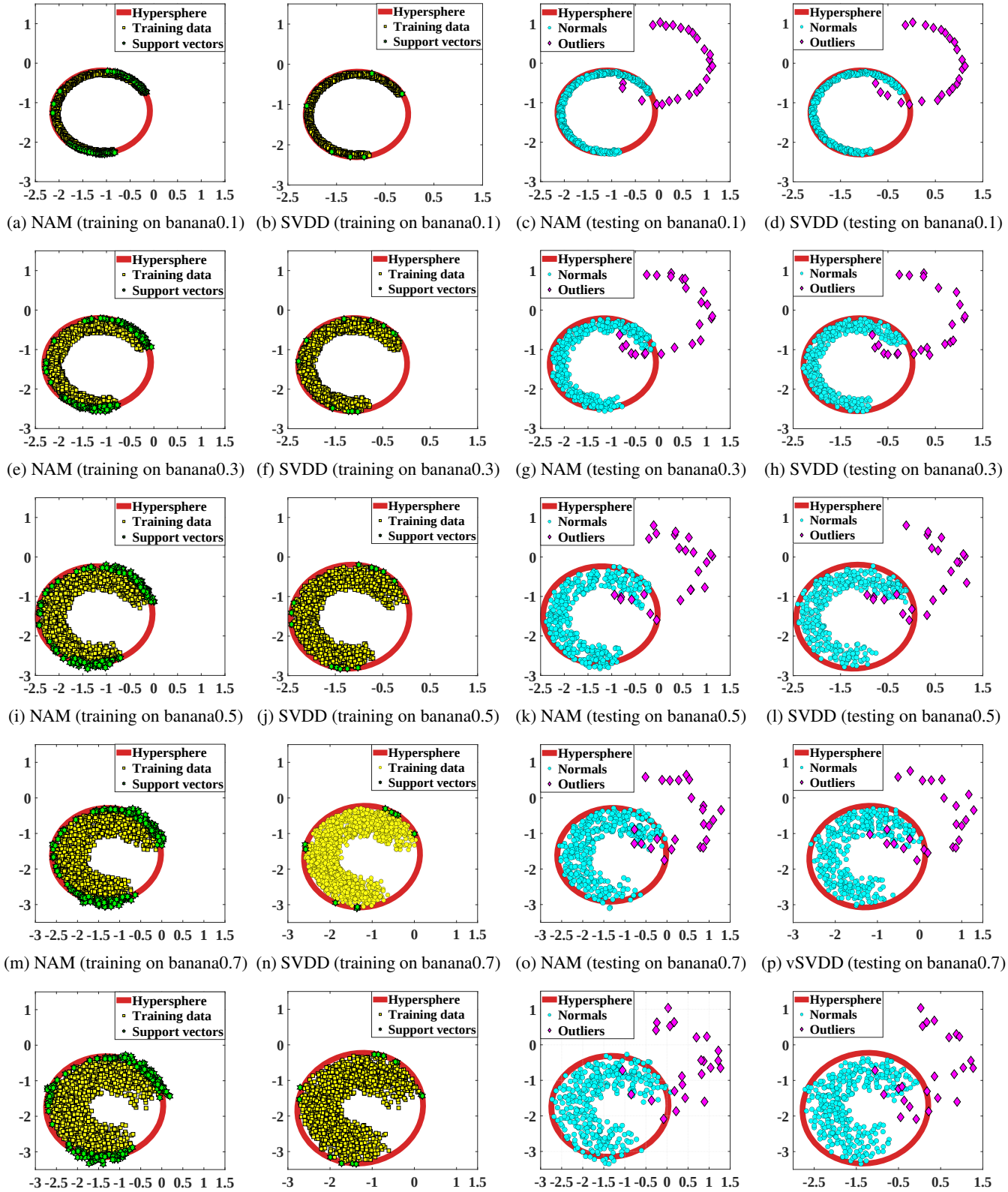
[Zhang *et al.*, 2022] Fengbin Zhang, Haoyi Fan, Ruidong Wang, Zuoyong Li, and Tiancai Liang. Deep dual support vector data description for anomaly detection on attributed networks. *Int. J. Intell. Syst.*, 37(2):1509–1528, 2022.

[Zhong *et al.*, 2022] Guangzheng Zhong, Yanshan Xiao, Bo Liu, Liang Zhao, and Xiangjun Kong. Pinball loss support vector data description for outlier detection. *Appl. Intell.*, 52(14):16940–16961, 2022.

[Zhou and Paffenroth, 2017] Chong Zhou and Randy C. Paffenroth. Anomaly detection with robust deep autoencoders. In *Proceedings of the 2017 International Conference on Knowledge Discovery and Data Mining, SIGKDD*, pages 665–674, 2017.

[Zhou *et al.*, 2022] Yingjie Zhou, Xucheng Song, Yanru Zhang, Fanxing Liu, Ce Zhu, and Lingqiao Liu. Feature encoding with autoencoders for weakly supervised anomaly detection. *IEEE Trans. Neural Networks Learn. Syst.*, 33(6):2454–2465, 2022.

[Zong *et al.*, 2018] Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Dae-ki Cho, and Haifeng Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *Proceedings of the 2018 International Conference on Learning Representations, ICLR*, 2018.



(a) NAM (training on banana0.1) (b) SVDD (training on banana0.1) (c) NAM (testing on banana0.1) (d) SVDD (testing on banana0.1)

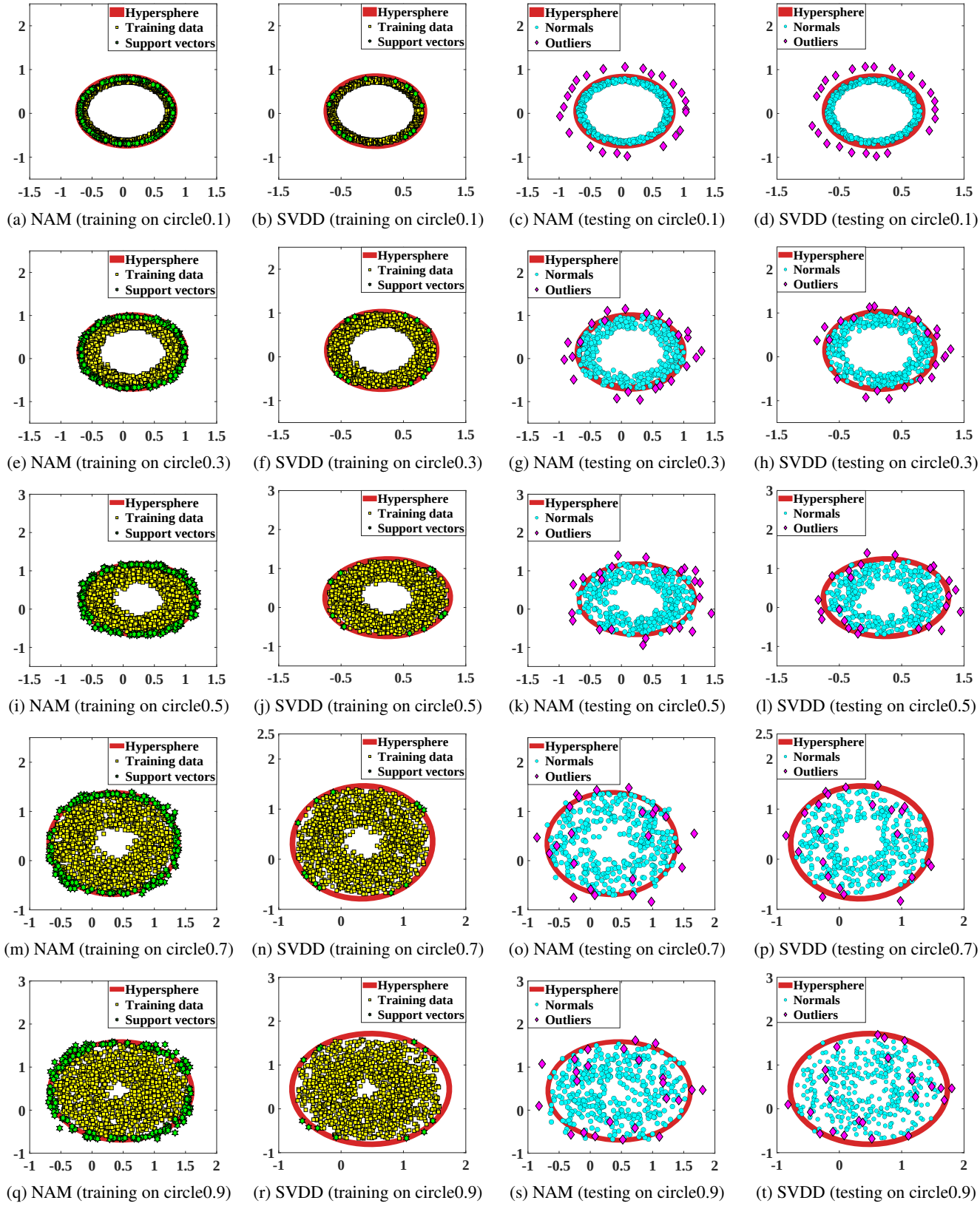
(e) NAM (training on banana0.3) (f) SVDD (training on banana0.3) (g) NAM (testing on banana0.3) (h) SVDD (testing on banana0.3)

(i) NAM (training on banana0.5) (j) SVDD (training on banana0.5) (k) NAM (testing on banana0.5) (l) SVDD (testing on banana0.5)

(m) NAM (training on banana0.7) (n) SVDD (training on banana0.7) (o) NAM (testing on banana0.7) (p) vSVDD (testing on banana0.7)

(q) NAM (training on banana0.9) (r) SVDD (training on banana0.9) (s) NAM (testing on banana0.9) (t) SVDD (testing on banana0.9)

Figure 3: AUC of SVDD and NAM on five datasets.



(a) NAM (training on circle0.1) (b) SVDD (training on circle0.1) (c) NAM (testing on circle0.1) (d) SVDD (testing on circle0.1)

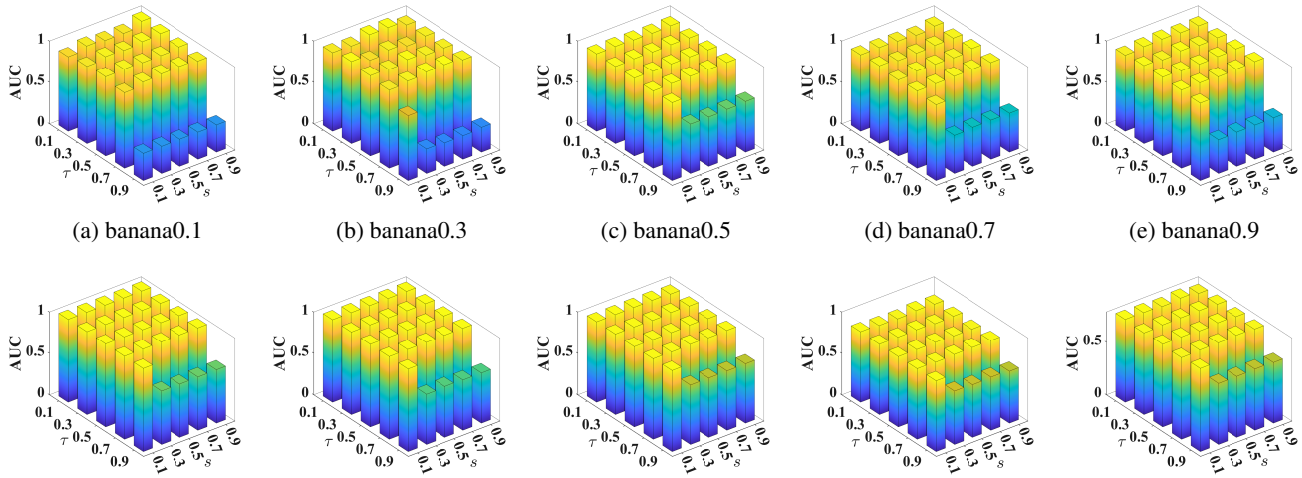
(e) NAM (training on circle0.3) (f) SVDD (training on circle0.3) (g) NAM (testing on circle0.3) (h) SVDD (testing on circle0.3)

(i) NAM (training on circle0.5) (j) SVDD (training on circle0.5) (k) NAM (testing on circle0.5) (l) SVDD (testing on circle0.5)

(m) NAM (training on circle0.7) (n) SVDD (training on circle0.7) (o) NAM (testing on circle0.7) (p) SVDD (testing on circle0.7)

(q) NAM (training on circle0.9) (r) SVDD (training on circle0.9) (s) NAM (testing on circle0.9) (t) SVDD (testing on circle0.9)

Figure 4: AUC of SVDD and NAM on five datasets.



(a) banana0.1 (b) banana0.3 (c) banana0.5 (d) banana0.7 (e) banana0.9

(f) circle0.1 (g) circle0.3 (h) circle0.5 (i) circle0.7 (j) circle0.9

Figure 6: AUC of NAM with different parameters on 10 synthetic datasets.