Received 7 August 2023, accepted 19 September 2023, date of publication 18 October 2023, date of current version 27 October 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3325734

RESEARCH ARTICLE

A Complete Deep Support Vector Data Description for One Class Learning

RENXUE JIANG, ZHIJI YANG, (Member, IEEE), AND JIANHUA ZHAO, (Senior Member, IEEE)
School of Statistics and Mathematics, Yunnan University of Finance and Economics, Kunming, Yunnan 650221, China

Corresponding author: Zhiji Yang (yangzhiji@ynufe.edu.cn)

This work was supported in part by the Graduate Innovation Fund Project of the Yunnan University of Finance and Economics under Grant 2021YUFEYC081, in part by the Scientific Research Fund Project of Yunnan Provincial Department of Education under Grant 2022Y546, in part by the Scientific Research Fund Project of Yunnan Provincial Department of Science and Technology under Grant 202001AU070064, and in part by the National Natural Science Foundation of China under Grant 62006206 and Grant 12161089.

ABSTRACT In recent years, Deep Support Vector Data Description (Deep SVDD) has emerged as a leading method in the field of anomaly detection. However, inaccuracies in parameter solving have been identified as a limitation of this approach, which negatively affects its accuracy and efficiency. To address this issue, we propose a new method, called Complete Deep Support Vector Data Description (CD-SVDD). Our CD-SVDD is constructed with a traditional deep neural network and utilizes a modified SVDD as its last layer. Its parameters are solved by an alternate iteration algorithm that ensures both high precision and fast convergence of solutions. By keeping the network weights fixed, we solve the center and radius of the modified SVDD based on its convex dual optimization problem. With the exact center and radius, we then update the parameters of the neural network by backpropagation. Compared to the existing deep SVDD, all parameters of our method are precisely solved. So, our method is defined to be "complete". This approach enables us to maintain the ν-property found in shallow SVDD, which is beneficial for parameter selection and model interpretability. To evaluate the performance of CD-SVDD, we conducted extensive numerical experiments with five existing methods on two image datasets, CIFAR-10 and CIFAR-100, as well as five recorded benchmark datasets. Our results demonstrate that CD-SVDD achieves superior accuracy and efficiency in the detection of anomalies.

INDEX TERMS Anomaly detection, deep learning, support vector data description, strong duality.

I. INTRODUCTION

Anomaly detection is a widely research field in machine learning and data mining, with the aim of identifying data that deviates from most instances. It includes point anomaly, contextual anomaly, and collective anomaly [1], [2], [3]. Point anomaly, in particular, has been extensively researched and can be categorized into classification-based, clustering-based, and nearest neighbor-based approaches [1]. Among these methods, one-class classification models are commonly used and have shown admirable performance in various applications.

One-class classification models are trained using normal data to detect abnormal instances in prediction [4]. The classical approach is the one-class support vector machine (OCSVM) [5], [6], which assumes the origin is an abnormal point and learns a hyperplane to separate normal data from it. Although OCSVM has achieved great performance in various applications, it is limited by the use of a hyperplane to separate the data. Support vector data description (SVDD) [7], as a successful extension of OCSVM, separates normal and abnormal data by learning a hypersphere instead of a hyperplane. SVDD is more flexible and shows outstanding prediction performance.

To enhance the handling of high-dimensional data, some traditional dimension reduction methods have been employed. For example, in [8], the author utilizes principal component analysis (PCA) to reduce the dimension in image anomaly detection. In [9], Shravan et al. propose a document classifier based on PCA and OCSVM. In [10], Shen et al.

The associate editor coordinating the review of this manuscript and approving it for publication was Davide Patti.

employ PCA and SVDD for non-linear process monitoring. However, despite their convenience, these dimension reduction methods may significantly impact prediction accuracy. Moreover, traditional kernel techniques may have limited adaptability when dealing with complex data structures.

In recent years, neural networks have been employed in anomaly detection due to their robust data representation capabilities [11], [12], [13], [14], [15]. The main research involves two aspects.

1) Neural networks are often used for data preprocessing, followed by training of the anomaly detection model. For example, Alfeo et al. use an autoencoder for data dimension reduction before training the anomaly detection model in [16]. In [17], Wang et al. propose an unsupervised deep learning method based on an autoencoder combined with OCSVM for anomaly detection. These approaches are referred to as mixed models where the two stages are carried out separately. However, similar to traditional dimension reduction techniques, differences between normal and abnormal data are not directly detected in data preprocessing. As a result, the performance of anomaly detection cannot be guaranteed.
2) The traditional anomaly detection model has been extended to the deep learning framework, where neural networks and traditional models are often trained alternately. Ruff et al. proposed a deep SVDD by extending ν-SVDD to the deep learning framework in [18]. Similarly, in [19], a fully deep model, called a one-class neural network (OCNN), was proposed by extending OCSVM. These methods aim to improve the performance of anomaly detection by incorporating deep learning techniques.

The fully deep models mentioned above generally perform better in predicting anomalies on large and complex datasets. However, the deep SVDD training process has a limitation where the center c of the hypersphere is manually fixed and cannot be updated, negatively affecting the prediction performance. Moreover, the square of the hypersphere radius R^2 is substituted with the quantile of the square distance from c to the mapped samples, which lacks a theoretical foundation and hinders the convergence speed and prediction accuracy of deep SVDD.

To address the issue of imprecise solutions and to enhance model interpretability, we introduce a novel approach called the Complete Deep Support Vector Data Description (CD-SVDD). We also demonstrated the theoretical property of parameter ν in our CD-SVDD and propose an efficient algorithm for implementing CD-SVDD. Compared to existing model, all parameters of our model is solved precisely. Therefore, we defined it to be "complete".

The main contributions of this paper are summarized as follows.

1) We propose the CD-SVDD. It can be well adapted to the situation where the dimension of data is high and the distribution is complex. The CD-SVDD has great model properties and achieves higher prediction accuracy.
2) We further develop an efficient joint alternate algorithm for CD-SVDD, based on the strong duality of the optimization and backpropagation method. This algorithm is more efficient than the original algorithm and has a faster convergence speed.
3) We demonstrate the ν-property of CD-SVDD, which provides a valuable guidance for parameter selection and enhances the interpretability of the model.

The rest of this paper is outlined as follows. Section II briefly introduces the related work. In Section III, CD-SVDD is proposed and the corresponding efficient algorithm is developed. Then, we demonstrate the relevant ν-property. Then, in Section IV, abundant experiments are conducted to verify the validity of our method. Section V gives the conclusion.

II. RELATED WORK

In this section, we review the classical SVDD [7] and the deep SVDD [18].

A. SUPPORT VECTOR DATA DESCRIPTION

Let $\mathcal{X}$ be the input space. $X = \{x_1, x_2, \ldots, x_l\}$ is a training set that is drawn independently from $\mathcal{X}$. Let $\mathcal{H}$ be a reproducing kernel Hilbert space (RKHS) associated to a Mercer kernel $K : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ that is continuous, symmetric, and positive semidefinite [20]. Let $\phi : \mathcal{X} \mapsto \mathcal{H}$ be the associated feature map that satisfies $K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$ for all $x_i, x_j \in \mathcal{X}$.

SVDD [7] finds a hypersphere that contains all normal data as the separation boundary. It can be summarized as the following optimization problem:

$$\begin{aligned} \min_{R,c,\xi} \ & R^2 + C \sum_{i=1}^{l} \xi_i \\ \text{subject to} \ & \|\phi(x_i) - c\|^2 \leq R^2 + \xi_i, \\ & \xi_i \geq 0, i = 1, \ldots, l. \end{aligned} \tag{1}$$

Here, R is the radius of the hypersphere. c is the hypersphere center. ξ_i is the relaxation factor. The hyperparameter C is the penalty parameter. l is the sample size.

The solution of the primal problem (1) is usually obtained by solving the following dual problem (2).

$$\begin{aligned} \min_{\alpha} \ & \alpha^T Q \alpha - \sum_{i=1}^{l} \alpha_i Q_{ii} \\ \text{subject to} \ & 0 \leq \alpha \leq C, \\ & e^T \alpha = 1 \end{aligned} \tag{2}$$

where $Q_{ij} = K(x_i, x_j)$, α is the vector of Lagrangian multipliers, and e is a ones vector with appropriate dimensions.

Additionally, by replacing C of Eq. (1) with ν, ν-SVDD has been proposed in [18]. The corresponding optimization problem is formulated as:

$$\begin{aligned} \min_{R,c,\xi} \ & R^2 + \frac{1}{\nu l}\sum_{i=1}^{l}\xi_i \\ \text{subject to} \ & \|\phi(x_i) - c\| \le R^2 + \xi_i, \\ & \xi_i \ge 0, i = 1, \dots, l. \end{aligned} \tag{3}$$

Compared with C in SVDD, ν in ν-SVDD has a practical interpretation [21]. $\nu \in (0, 1]$ is proved to be not only the upper bound of the proportion of observations outside the hypersphere, but also the lower bound of the proportion of the support vectors [18].

B. DEEP SUPPORT VECTOR DATA DESCRIPTION

Deep SVDD is inspired by ν-SVDD [18], and uses a neural network instead of the traditional kernel function.

Soft-boundary deep SVDD is proposed to minimize the volume of the hypersphere that contains the outputs of the neural network. It can be formulated as:

$$\min_{W,R,c} R^2 + \frac{1}{\nu l}\sum_{i=1}^{l}\max\left\{0, \|\phi(x_i, W) - c\|^2 - R^2\right\} + \frac{\lambda}{2}\sum_{j=1}^{L}\left\|W^j\right\|_F^2. \tag{4}$$

Here, R is the hypersphere radius, c is the hypersphere center. W^j is the parameter matrix of the j-th layer neural network. L represents the number of layers of the neural network. $\phi(x_i, W)$ represents the outputs of the neural network with the input of the original data x_i. λ and ν are hyperparameters.

Furthermore, a simplified version of the deep SVDD, named one-class deep SVDD, is put forward as follows.

$$\min_{W,c} \frac{1}{l}\sum_{i=1}^{l}\|\phi(x_i, W) - c\|^2 + \frac{\lambda}{2}\sum_{j=1}^{L}\left\|W^j\right\|_F^2. \tag{5}$$

The above models extend the shallow SVDD to deep learning framework to improve the prediction performance. However, in their solution algorithms, the parameters R^2 and c are roughly calculated. Specifically, R^2 is just estimated by the quantile of the square distance from c to the data mapping result. This limits the performance of the deep SVDD.

III. A COMPLETE DEEP SUPPORT VECTOR DATA DESCRIPTION

As discussed in Section II-A, formulation (3) of ν-SVDD was proposed to improve the interpretability of the model. However, the non-convexity of the optimization problem with respect to R and the lack of guaranteed strong duality make it challenging to solve R^2 exactly [22]. In deep SVDD, the distance quantile is used as an approximate representation of R^2. However, in the deep-SVDD training process, the theoretical properties of ν in ν-SVDD cannot be guaranteed, which impedes the model's interpretability.

In this section, we propose the Complete Deep Support Vector Data Description(CD-SVDD). The parameters in CD-SVDD could be solved accurately, and it has greater theoretical properties.

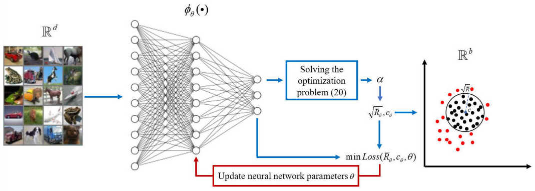

$\mathbb{R}^d$
$\phi_\theta(\cdot)$
Solving the optimization problem (20)
α
$\sqrt{\bar{R}_\theta}, c_\theta$
$\min Loss(\bar{R}_\theta, c_\theta, \theta)$
Update neural network parameters θ
$\mathbb{R}^b$
$\sqrt{\bar{R}}$
c

FIGURE 1. A schematic diagram of CD-SVDD. In solving the CD-SVDD, $\bar{R}_\theta$ and c_θ are represented by the output results of the neural network, which can be directly updated by the neural network parameter θ. That is why we define our method to be "complete".

Define the input space $\mathcal{X} \subseteq \mathbb{R}^d$ and the output space $\mathcal{F} \subseteq \mathbb{R}^b$. $\phi_\theta(\cdot) : \mathcal{X} \mapsto \mathcal{F}$ is a map from $\mathbb{R}^d$ to $\mathbb{R}^b$. Here, ϕ_θ is constructed by a neural network and θ is its corresponding weight parameters. l is the size of the training dataset $X = \{x_1, x_2, \dots, x_l\} \subseteq \mathcal{X}$. The aim of CD-SVDD is to jointly learn the neural network parameters θ together with minimizing the volume of a hypersphere containing the normal data. We give the formulation of CD-SVDD as follows.

$$\min_{\bar{R}_\theta, c_\theta, \theta} \bar{R}_\theta + \frac{1}{\nu l}\sum_{i=1}^{l}\max\left\{0, \|\phi_\theta(x_i) - c_\theta\|^2 - \bar{R}_\theta\right\} + \frac{\lambda}{2}\sum_{k=1}^{K}\left\|\theta^k\right\|. \tag{6}$$

This model constructs a hypersphere that contains normal data in the output space of the neural network. $\bar{R}_\theta$ is the square of the radius of the hypersphere. c_θ is the center of the hypersphere. $\phi_\theta(x_i)$ is the mapping result of x_i. The second term is a penalty for training data outside the hypersphere. The third term is the regularization for the parameters of the neural network. Here, it is assumed that the neural network has K layers and θ^k is the weights of the k-th layer. $\nu \in (0, 1]$ is a hyperparameter.

The schematic diagram in Fig. 1. illustrates the workflow of CD-SVDD. First, the original data in $\mathbb{R}^d$ are transformed into a low-dimensional space $\mathbb{R}^b$ by a neural network. Then, CD-SVDD finds the minimum volume hypersphere that encloses the normal data. During training, only normal data are expected to be mapped inside the hypersphere. In the prediction process, any data point outside the hypersphere is treated as an outlier.

Define the map of the neural network from the 1 first layer to the K -th layer as $\phi_{\theta^1}, \phi_{\theta^2}, \dots, \phi_{\theta^K}$. The feed-forward process of the CD-SVDD can be represented as

$$\begin{aligned} o &= \phi_{\theta^K}(\phi_{\theta^{K-1}}(\dots\phi_{\theta^1}(x))), \\ f(x) &= \operatorname{sgn}(\bar{R} - \|o - c\|^2). \end{aligned}$$

Here, a value of "+1" indicates normal, and a value of "−1" implies abnormal. That is, samples located inside the hypersphere are regarded as normal samples, while those located outside the hypersphere are abnormal samples.

In our proposed CD-SVDD, the optimization problem (6) will be solved exactly, giving the CD-SVDD greater accuracy and convergence speed. At the same time, the exact solution of the parameters of CD-SVDD could also provide better properties of the parameter ν, which makes CD-SVDD have good interpretability.

A. MODIFIED SVDD

The parameters of optimization problem (6) can be solved jointly and precisely. Firstly, assuming the parameters of neural network are fixed, $\bar{R}_\theta$, c_θ and ϕ_θ could be abbreviated as $\bar{R}$, c and ϕ. Then, (6) degenerates to following optimization problem:

$$\begin{aligned}\min_{\bar{R},c,\xi}\ & \bar{R}+\frac{1}{\nu l}\sum_{i=1}^{l}\xi_i\\ \text{subject to}\ & \|\phi(x_i)-c\|^2\le\bar{R}+\xi_i,\\ & \bar{R}\ge 0,\\ & \xi_i\ge 0, i=1,\ldots,l.\end{aligned}\tag{7}$$

Here, $\nu\in(0,1]$ is a hyperparameter. It is easy to demonstrate the convexity of the problem (7) which satisfies the Slater conditions. It often refers to strong duality [23]. This implies that the parameters $\bar{R}$ and c in the optimization problem (7) can be solved with precision. We call Eq. (7) a modified SVDD in this paper.

The modified SVDD holds the Theorem 1. It explains the relationship between the optimal solution of the optimization problem (7) and the parameter ν.

Theorem 1. *(a) For any $0<\nu<1$, the constraint $\bar{R}\ge 0$ in (7) is unnecessary. That is, without this constraint, any optimal solution still satisfies $\bar{R}\ge 0$.*

(b) If $\nu=1$, then at least one optimal solution has $\bar{R}=0$.

The proof is similar to Theorem 3 in [22], so we omit it here.

For (7), when $\nu=1$, there is at least one optimal solution $\bar{R}=0$ according to Theorem 1. Then, the problem (7) is equivalent to

$$\min_{c}\sum_{i=1}^{l}\|\phi(x_i)-c\|^2.\tag{8}$$

This case easily leads to a hypersphere collapse in neural networks [18]. Therefore, the setting of $\nu=1$ is not recommended.

When $\nu\in(0,1)$, the constraint $\bar{R}\ge 0$ in (7) will always be satisfied. The problem (7) could be written as

$$\begin{aligned}\min_{\bar{R},c,\xi}\ & \bar{R}+\frac{1}{\nu l}\sum_{i=1}^{l}\xi_i\\ \text{subject to}\ & \|\phi(x_i)-c\|^2\le\bar{R}+\xi_i,\\ & \xi_i\ge 0, i=1,\ldots,l.\end{aligned}\tag{9}$$

Note that if ν is set too close to 0, each ξ_i will tend to 0.

According to the optimization problem (7), $\bar{R}_\theta$, c_θ in the CD-SVDD can be accurately solved. However, to update the parameters of the neural network more accurately and efficiently, we solve $\bar{R}$, c in the optimization problem (7) via its corresponding dual problem [24].

The Lagrange function [23] of (9) can be written as follows.

$$\begin{aligned}L(\bar{R},c,\xi,\alpha,\beta)=\bar{R}+\frac{1}{\nu l}\sum_{i=1}^{l}\xi_i\\ +\sum_{i=1}^{l}\alpha_i\left(\|\phi(x_i)-c\|^2-\bar{R}-\xi_i\right)\\ -\sum_{i=1}^{l}\beta_i\xi_i.\end{aligned}\tag{10}$$

where α_i, β_i is the Lagrange multipliers. Let

$$\begin{aligned}\frac{\partial L}{\partial\bar{R}}&=1-\sum_{i=1}^{l}\alpha_i=0,\\ \frac{\partial L}{\partial c}&=-2\sum_{i=1}^{l}\alpha_i\left(\phi(x_i)-c\right)=0,\\ \frac{\partial L}{\partial\xi}&=\frac{1}{\nu l}e-\alpha-\beta=\mathbf{0},\end{aligned}$$

KKT conditions corresponding to (9) can be obtained as follows.

$$\|\phi(x_i)-c\|^2-\bar{R}-\xi_i\le 0, i=1,\ldots,l,\tag{11}$$

$$-\xi_i\le 0, i=1,\ldots,l,\tag{12}$$

$$\alpha_i\ge 0, i=1,\ldots,l,\tag{13}$$

$$\beta_i\ge 0, i=1,\ldots,l,\tag{14}$$

$$\beta_i\xi_i=0, i=1,\ldots,l,\tag{15}$$

$$\alpha_i\left(\|\phi(x_i)-c\|^2-\bar{R}-\xi_i\right)=0, i=1,\ldots,l,\tag{16}$$

$$\sum_{i=1}^{l}\alpha_i=1,\tag{17}$$

$$\sum_{i=1}^{l}\alpha_i\phi(x_i)=c,\tag{18}$$

$$\alpha+\beta=\frac{1}{\nu l}e.\tag{19}$$

The dual problem of (9) is achieved as follows.

$$\begin{aligned}\min_{\alpha}\ & \alpha^TQ\alpha-\sum_{i=1}^{l}\alpha_iQ_{ii}\\ \text{subject to}\ & 0\le\alpha_i\le\frac{1}{\nu l}, i=1,\ldots,l,\\ & e^T\alpha=1.\end{aligned}\tag{20}$$

Here, $Q_{ij}=\phi(x_i)*\phi(x_j)^T$.

Definition 1. *[25] Let $\alpha^* = (\alpha_1^*, \alpha_2^*, \ldots, \alpha_l^*)$ be the optimal solution of (20). The observation x_i is a support vector, if $\alpha_i^* \neq 0$; Otherwise, it is called a non-support vector.*

The optimal solution of center c^* can be obtained by the following formula:

$$c^* = \sum_{i=1}^{l} \alpha_i^* \phi(x_i). \tag{21}$$

Denoted $i_1, i_2, \ldots, i_p$ as the index that satisfies $0 < \alpha_i^* < 1/\nu l$ in α^*, then the optimal solution of radius square $\bar{R}^*$ is achieved by:

$$\bar{R}^* = \frac{1}{p} \sum_{j=1}^{p} \left\| \phi\left(x_{i_j}\right) - c^* \right\|^2. \tag{22}$$

B. DEEP LEARNING FOR θ

To update θ, we substitute the formulas (21) and (22) into (6). Then, (6) can be rewritten as:

$$\begin{aligned} \min_{\theta} \; & \frac{1}{p} \sum_{j=1}^{p} \left\| \phi_\theta\left(x_{i_j}\right) - \sum_{s=1}^{l} \alpha_s^* \phi_\theta\left(x_s\right) \right\|^2 + \frac{\lambda}{2} \sum_{k=1}^{K} \left\| \theta^k \right\| \\ & + \frac{1}{\nu l} \sum_{i=1}^{l} \max \left\{ 0, \left\| \phi_\theta\left(x_i\right) - \sum_{s=1}^{l} \alpha_s^* \phi_\theta\left(x_s\right) \right\|^2 \right. \\ & \left. - \frac{1}{p} \sum_{j=1}^{p} \left\| \phi_\theta\left(x_{i_j}\right) - \sum_{s=1}^{l} \alpha_s^* \phi_\theta\left(x_s\right) \right\|^2 \right\}. \end{aligned} \tag{23}$$

Formulation (23) serves as a loss function to update the parameters in the neural networks. Notably, $\bar{R}$ and c in (23) can be precisely solved through the dual problem (20). In comparison, deep SVDD in (4) and (5) only provides an approximate estimation of R^2 and c through quantiles. Hence, our proposed method offers stronger theoretical support and yields a more accurate solution. Additionally, $\bar{R}$ and c in (23) are represented by $\phi(x)$ based on the KKT conditions. This enables the direct optimization of the hypersphere volume as part of the loss function, resulting in a smaller hypersphere volume. Consequently, it becomes easier to separate abnormal samples from the hypersphere, further enhancing the accuracy of the model. Furthermore, the precise solution method accelerates convergence and enhances the computational speed of the model.

After the optimal parameters of CD-SVDD are obtained, for any sample x, we can define an anomaly score as follows.

$$s(x) = \left\| \phi_{\theta^*}(x) - c^* \right\|^2 - \bar{R}^*.$$

Then, we can give a prediction for x by the decision function:

$$f(x) = \mathrm{sgn}(s(x)).$$

C. A JOINT ALTERNATE ALGORITHM

Firstly, $\bar{R}$ and c are updated by solving the dual variable α in optimization problem (20) with the initial parameter θ of the neural network. After that, θ is updated according to (23). Parameters are trained alternately until the primal problem (6) converges. The pseudo code is given in Algorithm 1.

Algorithm 1 Joint Alternate Algorithm of CD-SVDD

```
Input: training data $X$, hyperparameter $\nu$
Output: $\bar{R}^*$, $c^*$, $\theta^*$
Initialize neural network parameter $\theta$
while objective function (6) does not converge do
    $\alpha^* \Leftarrow$ solving convex optimization problem (20) with $\phi_\theta(X)$
    $\theta \Leftarrow$ back propagation algorithm to minimize the loss (23)
end while
$\bar{R}^*$, $c^* \Leftarrow$ calculating by Eq. (21) and (22)
$\theta^* \Leftarrow \theta$
```

Algorithm 1 can be easily extended to the mini-batch case. Then, in each iteration, it just needs to solve a smaller optimization. Additionally, with the precisely calculating $\bar{R}^*$ and c^* in each batch, the whole algorithm can converge faster than deep SVDD.

D. ν-PROPERTY IN CD-SVDD

In this section, we demonstrate a great property of the parameter ν in CD-SVDD. For the sake of narrative, define the abstract sample sets A, B and C as follows.

$$A = \left\{ \phi_\theta(x) | \left\| \phi_\theta(x_i) - c_\theta \right\|^2 < \bar{R}_\theta, x \in X \right\},$$

$$B = \left\{ \phi_\theta(x) | \left\| \phi_\theta(x_i) - c_\theta \right\|^2 = \bar{R}_\theta, x \in X \right\},$$

$$C = \left\{ \phi_\theta(x) | \left\| \phi_\theta(x_i) - c_\theta \right\|^2 > \bar{R}_\theta, x \in X \right\}.$$

According to (13)-(17), following formula holds

$$\phi_\theta(x_i) \in A \Rightarrow \alpha_i^* = 0, \tag{24}$$

$$\phi_\theta(x_i) \in B \Rightarrow 0 < \alpha_i^* < \frac{1}{\nu l}, \tag{25}$$

$$\phi_\theta(x_i) \in C \Rightarrow \alpha_i^* = \frac{1}{\nu l}. \tag{26}$$

Then, the following ν-property can be proved.

Theorem 2. *For CD-SVDD, $\nu \in (0, 1)$ is not only the upper bound of the proportion of the mapped observations outside the hypersphere, but also the lower bound of the proportion of the support vectors.*

Proof: When $\nu \in (0, 1)$, according to (17) and (26), we have

$$|B| \frac{1}{\nu l} \leq \sum_{i=1}^{l} \alpha_i.$$

Besides, from (17), (25) and (26), we obtain

$$|B \cup C| \frac{1}{\nu l} \geq \sum_{i=1}^{l} \alpha_i.$$

Therefore, we have

$$|B|\frac{1}{\nu l} \leq \sum_{i=1}^{l} \alpha_i \leq |B \cup C|\frac{1}{\nu l}.$$

Namely,

$$|B|\frac{1}{l} \leq \nu \leq |B \cup C|\frac{1}{l}.$$

■

When min-batch strategy is employed, the result similar to Theorem 2 can be derived.

The ν-property makes CD-SVDD have greater interpretability, which is helpful to guide the parameter selection in the model training.

IV. COMPARISONS WITH OTHER EXISTING METHODS

In this section, we give a comparative discussion between CD-SVDD and other existing methods, including classical neural networks and anomaly detection methods.

A. COMPARISON WITH EXISTING CLASSICAL NEURAL NETWORK METHODS

The hidden layer structure of CD-SVDD inherits traditional neural networks. For instance, it could be made up of convolution and pooling layers. Most importantly, our CD-SVDD is designed for the specific task of anomaly detection, while traditional CNNs (convolutional neural networks) and GANs (generative adversary networks) are intended for ordinary classification tasks or generative tasks. One of the main constructive differences of our CD-SVDD is the loss function, which is tailored for anomaly detection. The CD-SVDD generates a hypersphere in the mapped feature space as an anomaly detector.

B. COMPARISON WITH EXISTING ANOMALY DETECTION APPROACHES

The comparisons of CD-SVDD with shallow one-class methods, mixed deep anomaly detection approaches, and full deep methods are discussed.

1) The shallow one-class methods, such as OCSVM and SVDD, use a kernel function to map the original data into high-dimensional space [5], [7]. However, this approach has weaker adaptability, especially when faced with complex data distributions. On the contrary, CD-SVDD learns a neural network to map the original data into a high-dimensional space, thereby improving model performance. The data representation ability of neural networks is more powerful than that of kernel functions, which results in CD-SVDD generally having better prediction performance than shallow models.

2) In mixed deep anomaly detection approaches, the neural network training process and anomaly detection are usually carried out independently. This limits the prediction performance. On the contrary, CD-SVDD trains the parameters of neural network and modified SVDD together, which is more conducive to improve prediction performance.

3) The existing deep SVDD model approximates some of its parameters using a quantile approximation, which may lead to imprecise results. In contrast, our CD-SVDD uses an alternative iteration training strategy. First, the parameters of the modified SVDD are obtained through dual optimization. Then, the parameters of the neural network are updated using backpropagation. This allows for precise updates to the model parameters during training, resulting in higher accuracy. The training process is also faster as the loss function converges more quickly. Furthermore, the parameter ν in CD-SVDD has better interpretability, which can provide useful guidance for parameter selection.

TABLE 1. Statistics of two image datasets.

Dataset	# Categories	# Features	Sample Size	
			Training	Test
CIFAR-10	10	3×32×32	50000	10000
CIFAR-100	100	3×32×32	50000	10000

V. NUMERICAL EXPERIMENTS

A. EXPERIMENTAL SETUP

To verify the advantages of CD-SVDD, numerical experiments are conducted on two image datasets, i.e., CIFAR-10, CIFAR-100,[1] and five recorded benchmark datasets, i.e., a9a [26], codrna [27], epileptic [28], htru2 [29] and ijcnn1 [30] from UCI Machine Learning Repository[2] and LIBSVM.[3] Their statistics are given in Tables 1 and 2, respectively. Additionally, we verify the ν-property in CD-SVDD on five recorded benchmark datasets.

CIFAR-10 and CIFAR-100 datasets contain rich physical images [31]. Some examples are shown in Fig. 2. These two datasets were provided with specific training and division of test sets. For the other five benchmark datasets, we randomly take 80% for training, and the remaining 20% for testing. In particular, one-class methods are just trained with one-class samples. For multiclass data, we take turns to use one of the categories of training samples to build the models.

All experiments are implemented by Python 3.10 on Windows 11 running on a PC with system configuration Intel Core i5-1140 CPU 2.70GHz with 16GB of RAM.

In the performance experiment, we compare CD-SVDD with isolated forest (IF), OCSVM, ν -SVDD, soft-boundary deep SVDD and one-class deep SVDD.

1) **IF** [32] is an anomaly detection method. It judges normal and abnormal data according to the average path lengths needed to "isolate" a data point. It holds that abnormal data

[1] http://www.cs.toronto.edu/ kriz/cifar.html

[2] https://archive.ics.uci.edu/ml/datasets

[3] https://www.csie.ntu.edu.tw/ cjlin/libsvmtools/datasets/binary.html

TABLE 2. Statistics of five recorded benchmark datasets.

Dataset	# Categories	# Features	Class	Sample Size	
				Training	Test
a9a	2	132	0	19776	12785
			1	6273	26288
codrna	2	8	0	31752	27783
			1	15876	43659
epileptic	5	178	0	1840	9660
			1	1840	9660
			2	1840	9660
			3	1840	9660
			4	1840	9660
htru2	2	8	0	13007	4891
			1	1311	16587
ijcnn1	2	22	0	36110	13880
			1	3882	46108

FIGURE 2. Some examples of image datasets. The left is from CIFAR-10. The right is from CIFAR-100.

can usually be "isolated" by short average path lengths on the trees.

2) **OCSVM** [5] is a one-class classifier. It assumes that the origin is an outlier and searches for a hyperplane farthest from the outlier as the separation boundary. The non-linear kernel is introduced to improve the model accuracy.

3) ν**-SVDD** [7] is a one-class classification. It separates normal and anormal data by learning a hypersphere that contains all normal data.

4) **Soft-boundary Deep SVDD** [18] is a fully deep anomaly detection model proposed based on ν-SVDD. A neural network is used as a mapping tool to train a hypersphere with the smallest volume.

5) **One-class Deep SVDD** [18] is a simplified version of the soft-boundary deep SVDD. Its objective function is changed to minimize the average distance between normal samples and its center.

In the experiments, IF is implemented by the "ensemble.IsolationForest" class in the sklearn package. Both OCSVM and ν-SVDD adopt the Gaussian kernel function [33] and the best hyperparameter is selected by grid search. σ is from the set $\{2^i | i = -7, -6.5, \ldots, 7\}$ and ν is from the set $\{0.01, 0.02, \ldots, 0.99\}$. For deep learning methods, that is, soft-boundary deep SVDD, one-class deep SVDD, and CD-SVDD, we fix ν=0.1 referring to [18] and the batch size is 200. On the CIFAR-10 and CIFAR-100 datasets, the architectures of their neural networks are CIFAR10-LeNet [18]. On the other five benchmark datasets, their architectures are fully connected feedforward neural networks. And we initialize network weights by uniform Glorot weights [34]. They are implemented through the torch package. The convex optimization problem involved in CD-SVDD is solved by the block coordinate descent method [35]. In order to verify the ν-property in CD-SVDD, we obtain the relationship between the CD-SVDD training error ratio, support vector ratio, and ν, where ν takes the value in $\{0.05, 0.1, \ldots, 0.95\}$.

Taking into account the case of imbalanced test samples (one of the classes is regarded normal class, and the others abnormal), the value of AUC (area under curve) under optimal parameters is used to evaluate the prediction performance of each model [36]. Furthermore, we calculate the F1 scores under the optimal parameters on the five recorded benchmark datasets.

B. RESULTS OF PREDICTION AND EFFICIENCY

To evaluate the accuracy and effectiveness of our proposed CD-SVDD method, we compare it with five other methods on two image datasets and five benchmark datasets.

1) **CIFAR-10.** It consists of images of common vehicles and animals. As shown in Table 3, CD-SVDD achieves higher AUC values in comparison to other models in 8 of 10 categories, which means that the model has greater accuracy. This suggests that CD-SVDD can more accurately identify common vehicles and animals. OCSVM and SVDD follow it.

Besides, the training iterations and computational time of three deep models, i.e., soft-boundary deep SVDD, one-class deep SVDD and CD-SVDD, are shown in Table 4. "Iterations" corresponds to the iterations of updating the model parameters, and "Time" represents the total training time until the model reaches the optimal AUC value. "DP" represents the time to solve the dual problem, and "Network" denotes the time to update the parameters of the neural network. It implies that CD-SVDD achieves the least iterations for 4 categories, and the shortest time for 5 categories. It can also be found that the computational time of CD-SVDD is shorter even with large iterations. This indicates that CD-SVDD has advantages in computation speed compared to the other two models.

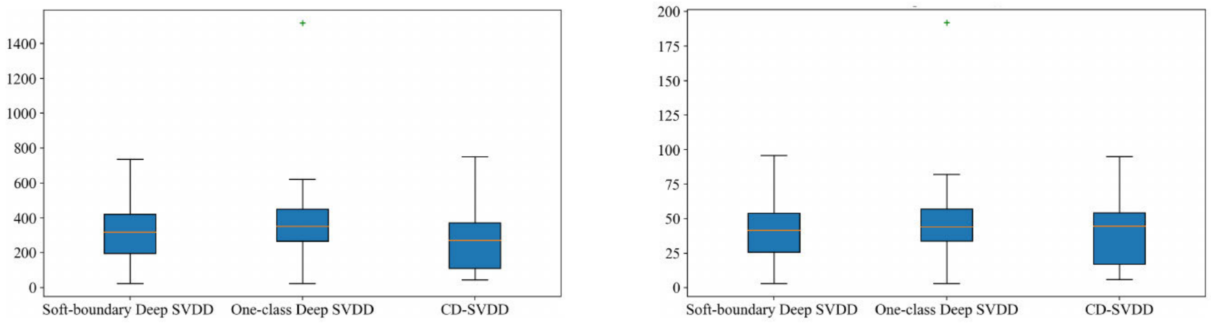
2) **CIFAR-100.** It is comprised of images of animals, plants, and household products, providing an extremely diverse range of objects for classification. As shown in Table 5, CD-SVDD outperforms other models in 90 of 100 categories in terms of optimal AUC values, indicating its higher accuracy in identifying anomalies. Moreover, based on the AVERAGE result, CD-SVDD significantly outperforms other models. These results indicate that CD-SVDD can be effectively applied to real-world anomaly detection tasks. The corresponding iterations and the results of the computational time are shown in Table 6. CD-SVDD achieves the lowest number of iterations for 45 categories and has the shortest computational time for 54 categories. Additionally, we provide a box plot in Fig. 3 to observe the overall computational performance in all 100 categories of

TABLE 3. The optimal AUC values (in Percentage) of six methods on CIFAR-10 dataset.

Normal Class	IF	OCSVM	SVDD	Soft-boundary Deep SVDD	One-class Deep SVDD	CD-SVDD
0	66.733	65.223	70.156	64.157	58.338	**74.853**
1	44.628	41.009	51.100	51.865	54.672	**79.937**
2	65.085	65.074	**68.820**	64.573	65.871	66.781
3	50.417	49.907	52.025	54.500	55.132	**59.611**
4	74.902	75.042	**77.302**	76.066	76.336	71.752
5	52.291	51.356	50.822	56.200	56.894	**63.282**
6	71.355	71.591	74.056	75.543	75.639	**77.453**
7	54.308	51.291	52.207	59.657	56.379	**66.745**
8	71.589	67.661	70.500	62.183	68.259	**76.441**
9	55.351	49.144	52.432	56.171	54.970	**79.172**
AVERAGE	60.666	58.730	56.883	62.092	62.249	**71.603**
Win/Draw/Loss	8/0/2	8/0/2	8/0/2	9/0/1	9/0/1	

TABLE 4. Computational iterations and time (in Seconds) of three deep models on CIFAR-10 dataset.

Normal Class	Soft-boundary Deep SVDD		One-class Deep SVDD		CD-SVDD			
	Iterations	Time (s)	Iterations	Time (s)	Iterations	Time (s)	DP (s)	Network (s)
0	32	252.570	35	308.683	**24**	**127.415**	0.316	127.099
1	89	695.009	**27**	208.069	31	**150.475**	0.396	150.079
2	62	466.353	**51**	**386.757**	207	1406.150	2.689	1403.461
3	**27**	**215.989**	467	3646.297	253	1760.750	3.300	1757.450
4	236	959.792	**69**	**545.986**	123	847.495	1.604	845.891
5	**16**	**118.008**	27	205.710	37	262.550	0.479	262.071
6	226	1671.144	226	1687.093	**49**	**317.277**	0.631	316.646
7	43	218.586	42	321.493	**11**	**68.962**	0.138	68.824
8	**29**	**249.317**	34	294.875	35	262.017	0.454	261.563
9	**16**	119.506	53	387.662	**16**	**103.893**	0.206	103.687
AVERAGE	**77.600**	**496.627**	103.100	799.263	78.600	530.698	1.021	529.677
Win/Draw/Loss	5/1/4	5/0/5	6/0/4	7/0/3				



(a) Iterations

(b) Time (s)

FIGURE 3. Iterations and time (in seconds) of three deep models on CIFAR-100 dataset.

CIFAR-100. The quartiles of iterations and corresponding time for CD-SVDD are significantly smaller than those for the other two deep models. This suggests that CD-SVDD can converge to the optimal AUC value more quickly, demonstrating the effectiveness of using optimization methods to solve the problem (9) and minimize the objective function (6). The outliers observed in the one-class deep SVDD solution process also suggest that CD-SVDD has a more stable solving process.

3) **Five recorded benchmark datasets.** We verify the advantages of our CD-SVDD in terms of prediction accuracy and computational efficiency. For prediction accuracy, we compare the optimal AUC value, average AUC value, and F1 score of each model on these datasets. Additionally,

TABLE 5. The optimal AUC values (in percentage) of six models on CIFAR-100.

Normal Class	IF	OCSVM	SVDD	Soft-boundary Deep SVDD	One-class Deep SVDD	CD-SVDD
0	74.248	56.793	74.985	54.251	64.991	**78.039**
1	73.031	68.116	66.822	62.753	68.229	**78.016**
2	55.272	53.449	55.423	60.366	58.166	**72.178**
3	64.300	62.458	62.323	64.083	66.183	**72.722**
4	74.722	74.430	73.096	67.361	71.796	**75.568**
5	60.157	58.538	58.202	59.836	57.322	**65.516**
6	61.665	57.804	56.893	64.980	64.927	**70.053**
7	71.667	70.820	68.347	70.594	73.592	**77.451**
8	67.741	66.535	68.349	61.884	62.828	**73.023**
9	61.245	55.575	64.667	54.681	62.161	**66.648**
10	52.084	50.754	57.152	57.511	59.984	**60.838**
11	47.188	45.918	45.701	54.953	53.331	**68.799**
12	72.397	68.864	70.399	65.259	72.245	**77.132**
13	51.504	48.140	49.840	65.039	60.383	**72.068**
14	59.762	56.547	57.208	62.864	65.137	**76.342**
15	62.425	60.147	59.838	62.237	58.734	**62.476**
16	58.974	54.077	54.616	62.769	62.957	**64.927**
17	85.357	83.005	83.119	71.654	62.559	**88.313**
18	74.862	75.043	75.724	71.070	**76.193**	69.140
19	61.254	57.966	58.421	63.481	**64.480**	63.802
20	81.197	69.155	76.175	42.229	62.510	**83.587**
21	70.958	63.482	64.948	69.806	71.700	**79.185**
22	48.479	47.231	53.495	59.122	62.062	**66.829**
23	80.082	79.658	80.766	53.620	82.214	**91.267**
24	**89.509**	83.665	84.842	48.807	57.832	88.615
25	52.276	49.845	53.009	51.445	51.404	**62.277**
26	56.700	57.463	60.055	67.565	64.959	**70.228**
27	75.795	75.704	75.847	63.760	69.910	**77.332**
28	62.290	58.216	74.291	62.092	62.188	**78.833**
29	56.477	52.890	57.101	47.667	56.244	**72.273**
30	**86.162**	84.879	83.169	60.501	66.500	84.950
31	66.857	66.181	67.086	68.702	67.352	**71.500**
32	59.885	50.031	50.841	58.855	59.095	**64.293**
33	**84.765**	83.093	81.824	68.800	76.800	82.883
34	70.125	70.196	73.317	64.365	70.403	**73.819**
35	47.420	44.973	49.975	61.322	58.061	**70.065**
36	72.448	70.627	69.593	64.489	61.789	**77.393**
37	66.113	61.759	63.259	65.534	59.430	**69.871**
38	76.458	76.360	**76.527**	60.780	73.292	72.261
39	58.290	55.709	67.420	70.339	61.524	**73.502**
40	51.085	47.933	62.340	53.882	58.543	**65.223**
41	74.098	62.081	69.153	42.558	43.733	**77.958**
42	74.574	72.493	**77.610**	63.300	71.255	74.632
43	77.605	76.231	76.445	68.298	**78.042**	72.335
44	64.201	64.582	67.301	59.350	62.545	**70.395**
45	54.253	52.728	54.084	58.583	61.967	**69.238**
46	45.723	41.915	49.985	55.481	53.966	**70.910**
47	81.518	75.394	78.494	71.596	73.940	**86.686**
48	52.420	50.946	51.684	69.080	64.694	**79.629**
49	80.903	78.145	79.977	67.574	65.868	**85.758**
50	61.472	62.017	52.842	53.750	57.420	**73.236**
51	73.598	70.430	71.141	69.919	73.507	**78.601**
52	91.947	88.032	88.121	85.357	84.972	**94.354**
53	82.982	66.542	75.822	40.974	47.550	**93.593**
54	54.276	50.972	47.060	63.725	63.876	**78.500**
55	65.707	66.286	65.101	57.880	60.900	**68.513**
56	65.074	63.283	62.356	62.997	65.376	**72.930**
57	60.940	53.642	62.450	52.105	51.894	**71.339**
58	57.758	54.907	56.266	71.434	69.572	**75.659**
59	74.339	69.698	70.173	67.715	68.601	**82.649**
60	89.592	86.987	93.211	80.611	80.384	**96.909**
61	74.279	68.843	71.188	66.435	65.289	**76.642**
62	69.915	61.633	63.130	31.665	65.145	**84.256**
63	77.651	76.977	81.101	62.290	72.174	**81.609**
64	69.313	68.851	69.520	63.426	63.266	**71.519**
65	66.320	64.816	66.499	60.620	60.580	**67.951**
66	67.161	66.607	65.017	59.634	67.109	**67.438**
67	71.965	69.844	75.389	44.640	66.900	**79.402**
68	83.482	81.167	83.241	68.751	62.560	**91.396**
69	72.348	69.595	76.468	61.399	65.882	**79.584**
70	63.738	58.252	61.152	45.293	56.175	**77.748**
71	89.055	88.075	89.883	74.753	84.596	**90.855**
72	63.023	61.835	61.696	62.056	56.574	**64.749**
73	80.422	74.723	76.378	53.020	79.954	**84.961**
74	73.370	73.484	73.426	64.273	72.855	**77.905**
75	66.715	63.347	65.951	66.917	73.351	**75.406**
76	75.026	71.304	74.600	59.235	68.459	**84.222**
77	67.149	67.964	68.377	63.580	62.430	**70.198**
78	67.992	67.741	66.749	62.097	70.641	**71.781**
79	74.202	76.537	**81.787**	66.320	70.323	80.925
80	73.335	72.728	74.655	68.081	73.484	**76.926**
81	60.944	57.518	58.101	68.166	62.545	**69.637**
82	67.843	55.598	57.276	46.173	63.821	**79.631**
83	58.796	51.873	56.431	49.504	53.436	**83.918**
84	55.494	55.181	59.272	55.929	51.873	**62.430**
85	71.077	66.542	65.370	65.512	58.546	**81.179**
86	65.444	51.637	49.919	44.536	48.041	**73.200**
87	51.480	47.808	53.950	55.742	58.682	**60.255**
88	69.263	68.140	68.222	66.289	70.142	**70.709**
89	69.615	68.977	67.729	73.828	72.106	**79.660**
90	65.050	60.933	62.149	60.216	54.835	**78.521**
91	66.119	61.460	71.102	57.096	63.650	**77.453**
92	68.313	63.574	64.642	70.794	71.570	**80.662**
93	58.565	58.541	62.753	59.955	56.540	**67.681**
94	75.620	72.182	75.160	67.201	68.155	**80.197**
95	77.628	72.511	70.928	59.878	60.622	**83.583**
96	79.828	76.359	75.290	71.012	71.407	**83.151**
97	70.188	67.741	67.807	70.674	**71.463**	69.513
98	44.104	41.533	51.000	56.031	52.629	**66.776**
99	62.903	64.740	73.338	76.281	81.600	**84.126**
AVERAGE	67.119	64.485	66.567	61.629	64.576	**75.708**
Win/Draw/Loss	93/0/7	96/0/4	95/0/5	98/0/2	95/0/5	

we carried out a sign test and Friderman test on the experimental results. For computational efficiency, we compare computational time and iterations when the models achieve desirable and stable AUC values.

TABLE 6. Iterations and time (in seconds) of three deep models on CIFAR-100 dataset.

Normal Class	Soft-boundary Deep SVDD		One-class Deep SVDD		CD-SVDD			
	Iterations	Time (s)	Iterations	Time (s)	Iterations	Time (s)	DP (s)	Network (s)
AVERAGE	45.270	350.329	43.010	334.963	**41.410**	**276.790**	0.540	276.250
Win/Draw/Loss	54/5/41	50/3/47	66/0/34	64/0/36				

TABLE 7. The optimal AUC values (in percentage) of six models on five recorded benchmark datasets.

Dataset	Normal Class	IF	OCSVM	SVDD	Soft-boundary Deep SVDD	One-class Deep SVDD	CD-SVDD
a9a	0	64.568	60.540	61.855	64.284	67.939	**69.543**
	1	77.615	77.271	78.453	69.447	71.541	**81.221**
codrna	0	69.966	67.087	72.168	79.336	**83.818**	82.335
	1	65.745	61.752	**92.330**	85.892	84.166	88.326
epileptic	0	1.679	1.149	32.702	95.111	95.920	**97.343**
	1	61.553	60.938	**71.367**	65.818	68.908	68.700
	2	65.447	65.887	73.785	72.339	71.908	**74.435**
	3	48.519	47.918	49.973	60.222	59.924	**70.804**
	4	75.609	76.347	76.445	47.400	67.240	**76.926**
htru2	0	95.102	92.982	94.991	94.991	95.665	**97.056**
	1	92.451	88.645	97.419	95.208	94.683	**97.504**
ijcnn1	0	58.715	58.440	65.884	61.152	63.101	**69.989**
	1	58.039	59.865	**81.403**	61.116	71.362	64.916
AVERAGE		63.728	62.986	72.983	73.255	76.629	**79.931**
Win/Draw/Loss		13/0/0	13/0/0	10/0/3	13/0/0	10/0/3	

TABLE 8. The average AUC values (in percentage) of six models on five recorded benchmark datasets.

Dataset	Normal Class	IF	OCSVM	SVDD	Soft-boundary Deep SVDD	One-class Deep SVDD	CD-SVDD
a9a	0	62.772 ± 2.174	60.540	61.855	61.986 ± 2.290	64.525 ± 2.426	**66.662±2.312**
	1	77.711 ± 0.374	77.271	78.453	68.489 ± 0.906	70.941 ± 0.826	**80.683±0.431**
codrna	0	69.524 ± 0.366	67.087	72.168	77.801 ± 2.042	78.908 ± 3.489	**81.566±0.754**
	1	65.229 ± 0.459	61.752	**92.330**	85.296 ± 0.446	83.963 ± 0.203	85.220 ± 2.249
epileptic	0	1.488 ± 0.135	1.149	32.702	93.920 ± 1.074	93.809 ± 2.274	**96.636±0.501**
	1	61.465 ± 0.090	60.938	**71.367**	65.405 ± 0.480	67.919 ± 0.885	68.361 ± 0.240
	2	65.363 ± 0.073	65.887	73.785	71.048 ± 0.990	71.483 ± 0.314	**73.960±0.374**
	3	48.225 ± 0.210	47.918	49.973	59.539 ± 0.626	59.058 ± 0.645	**70.753±0.048**
	4	75.437 ± 0.124	76.347	76.445	46.800 ± 0.648	65.247 ± 1.650	**76.739±0.173**
htru2	0	95.080 ± 0.016	92.982	94.991	94.664 ± 0.273	95.227 ± 0.324	**96.190±0.649**
	1	91.336 ± 0.841	88.645	97.419	94.722 ± 0.365	93.020 ± 1.380	**97.461±0.036**
ijcnn1	0	58.161 ± 0.652	58.440	65.884	60.160 ± 0.752	62.095 ± 0.891	**66.147±0.819**
	1	56.678 ± 0.963	59.865	**81.403**	60.103 ± 0.743	65.978 ± 3.966	63.015 ± 1.955
Win/Draw/Loss		13/0/0	10/0/3	13/0/0	12/0/1	12/0/1	

Considering that the performance of neural structures is often sensitive to initial conditions, we repeat the training three times and compare the corresponding optimal and average AUC values, respectively. As shown in Table 7, in terms of optimal AUC value, CD-SVDD has outstanding performance in 9 of 13 categories. As shown in Table 8, in terms of the average AUC value, CD-SVDD performed well in 10 out of 13 categories. Since the training results of OCSVM and SVDD on the same data are stable, we just train these models once. For shallow models IF, OCSVM and SVDD, the class '0' of dataset codrna cannot be detected properly at all, which reflects the limitations of shallow models.

We conduct a sign test [37] to further verify the prediction performance of CD-SVDD according to optimal AUC value in Table 7. The results are shown in Table 9. The total number of sample categories is denoted as ”N”, while “Node” represents the number of categories where the efficiency of the two models is equal. “S^+” and “S^-” are the number of categories in which CD-SVDD is superior or inferior to the other model, respectively. “p” indicates the results of the sign test. As shown in Table 9, at a significance level of

TABLE 9. Sign test of model precision in seven datasets.

Model	N	Node	S^+	S^-	p
IF	123	0	114	9**	0.000
OCSVM	123	0	117	6**	0.000
SVDD	123	0	113	10**	0.000
Soft-boundary Deep SVDD	123	0	120	3**	0.000
One-class Deep SVDD	123	0	114	9**	0.000

- $*$ and $**$ represent significant at levels of 0.1 and 0.05, respectively.

TABLE 10. The mean rank of AUC value of six models on five recorded benchmark datasets.

Model	Maximum AUC Value	Average AUC Value
IF	2.38	2.31
OCSVM	1.54	1.62
SVDD	4.35	4.38
Soft-boundary Deep SVDD	3.19	3.23
One-class Deep SVDD	4	3.85
CD-SVDD	5.54	5.62

TABLE 11. The statistics of friderman test.

	Average AUC Value
N	13
Chi-Square	37.714**
df	5
p	0.000

- $*$ and $**$ represent significant at levels of 0.1 and 0.05, respectively.

TABLE 12. Sign test of model precision in seven datasets.

Model	N	Node	S^+	S^-	p
IF	13	0	13	0**	0.000
OCSVM	13	0	10	3**	0.046
SVDD	13	0	13	0**	0.000
Soft-boundary Deep SVDD	13	0	12	1**	0.002
One-class Deep SVDD	13	0	12	1**	0.002

- $*$ and $**$ represent significant at levels of 0.1 and 0.05, respectively.

0.05, CD-SVDD outperforms the other five models in terms of prediction performance. This implies that the predictive performance of CD-SVDD is effective.

The Friderman test is conducted according to average AUC value in Table 8. As shown in Table 10 and Table 11, there are obvious differences in the prediction accuracy of each model.

Subsequently, we conducted a sign test to assess the disparity in prediction accuracy between CD-SVDD and each model according to Table 8. Details of the results are presented in Table 12, all of which demonstrated statistical significance at a confidence level of 0.05. This implies that the superior prediction accuracy exhibited by CD-SVDD is universal.

The optimal F1 score for each model is shown in Table 13. Among the 13 categories, CD-SVDD is significantly superior to other models in 9 categories. The one-class deep SVDD follows it. This indicates that CD-SVDD not only achieved success on image datasets but also has excellent performance on recorded datasets. This demonstrates the universality of CD-SVDD for data types.

The iterations and time of soft-boundary deep SVDD, one-class deep SVDD, and CD-SVDD are shown in Table 14. CD-SVDD achieves the least iterations and the shortest time for 5 categories of benchmark datasets, which verifies the efficiency of our method.

We conduct a sign test [37] to further verify the computational efficiency of CD-SVDD. As shown in Table 15, at a significance level of 0.1, the convergence rate of CD-SVDD is significantly better than that of soft-boundary deep SVDD. At a significance level of 0.05, the computational efficiency of CD-SVDD is also significantly better than that of one-class deep SVDD. These results demonstrate the effectiveness of applying convex optimization methods to the solving process.

It is important to note that, based on the results presented in Table 4, Table 6, and Table 14, the time required to solve the convex optimization problem (20) in our CD-SVDD is very minimal and can be considered negligible. Nevertheless,

TABLE 13. The optimal F1 score (in percentage) of six models on five recorded benchmark datasets.

Dataset	Normal class	IF	OCSVM	SVDD	Soft-boundary Deep SVDD	One-class Deep SVDD	CD-SVDD
a9a	0	55.949	39.246	24.765	56.693	**60.614**	58.561
	1	12.528	6.792	16.440	12.547	20.784	**21.346**
codrna	0	45.754	28.338	44.468	57.316	**60.492**	57.553
	1	17.860	11.646	27.448	25.184	33.060	**38.609**
epileptic	0	8.656	61.431	1.832	48.657	63.086	**65.493**
	1	10.623	7.200	1.619	12.489	13.471	**14.272**
	2	12.125	6.998	5.380	13.254	14.637	**15.533**
	3	11.385	9.231	4.601	11.810	11.711	**16.060**
	4	15.902	6.200	0.000	14.056	13.771	**15.938**
htru2	0	91.739	50.365	19.557	93.571	**93.861**	93.109
	1	20.692	1.821	31.313	63.188	55.043	**77.722**
ijcnn1	0	77.695	54.144	47.512	79.261	**79.678**	79.176
	1	4.177	3.255	4.299	4.272	5.084	**5.483**
Win/Draw/Loss		13/0/0	13/0/0	13/0/0	13/0/0	9/0/4	

TABLE 14. Iterations and time (in seconds) of three deep models on five recorded benchmark datasets.

Dataset	Normal class	Soft-boundary Deep SVDD		One-class Deep SVDD		CD-SVDD			
		Iterations	Time (s)	Iterations	Time (s)	Iterations	Time (s)	DP (s)	Network (s)
a9a	0	33	3.621	645	69.038	**20**	**2.383**	0.372	2.011
	1	18	4.941	**11**	**2.887**	32	9.196	0.579	8.616
codrna	0	**298**	**61.527**	512	105.923	1994	456.221	35.732	420.489
	1	767	244.774	**278**	**90.293**	671	234.057	12.043	222.014
epileptic	0	95	11.213	83	9.671	**43**	**5.709**	0.726	4.983
	1	**58**	**6.483**	378	37.933	165	21.801	2.946	18.855
	2	**36**	**4.037**	45	4.654	69	8.942	1.242	7.700
	3	117	12.938	**49**	**5.488**	70	7.883	1.241	6.641
	4	170	21.355	594	69.604	**24**	**2.992**	0.420	2.572
htru2	0	283	11.174	1973	76.138	**106**	**5.799**	1.877	3.921
	1	**11**	**1.350**	14	1.801	24	3.568	0.430	3.138
ijcnn1	0	**162**	**17.435**	4792	712.441	216	28.237	3.891	24.345
	1	159	56.674	182	64.203	**19**	**6.792**	0.326	6.465
AVERAGE		**169.769**	**35.194**	735.077	96.160	265.615	61.045	4.756	56.289
Win/Draw/Loss		7/0/6	7/0/6	7/0/6	7/0/6				

TABLE 15. Sign test of model efficiency in seven datasets.

Model		N	Node	S^+	S^-	p
Soft-boundary Deep SVDD	Iterations	123	6	66	51*	0.098
	Time	123	3	62	58	0.392
One-class Deep SVDD	Iterations	123	6	73	44**	0.005
	Time	123	0	78	45**	0.002

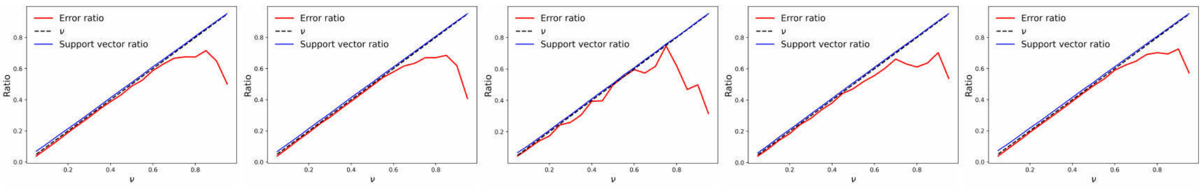
- $*$ and $**$ represent significant at levels of 0.1 and 0.05, respectively.

this optimization process plays a crucial role in our method by allowing for higher prediction performance and faster training speeds through an exact solution procedure. These findings further highlight the validity and effectiveness of utilizing convex optimization to calculate $\bar{R}$ and C in our CD-SVDD method.

C. RESULTS ON ν-PROPERTY OF CD-SVDD

We verify the ν-property of Theorem 2 using five recorded benchmark datasets. In the interest of brevity, we present results for only one class of training samples, labeled 0 for each dataset. The training error ratio and the support vector ratio curves, which change with the parameter ν, are shown in Fig. 4.

As the value of ν increases, the support vector ratio and training error ratio generally rise, indicating that ν controls the volume of the hypersphere. A larger value of ν may result in more training samples lying outside the hypersphere. It is obvious that the support vector ratio is never less than ν, and the error ratio is never greater than ν. This



(a) a9a (b) codrna (c) epileptic (d) htru2 (e) ijcnn1

FIGURE 4. The changing curves of training error ratio, support vector ratio with different values of ν.

observation confirms Theorem 2. Other datasets have similar conclusions.

VI. CONCLUSION

In this paper, we introduce the Complete Deep Support Vector Data Description (CD-SVDD) and propose an efficient solving algorithm that accurately computes each parameter using optimization methods with fast computational speed. By training the parameters of the neural network and the modified-SVDD jointly and updating them alternately, we achieve faster convergence of the objective function and higher prediction accuracy. Additionally, we demonstrate the ν-property of CD-SVDD, which not only sets the upper bound of the proportion of mapped observations outside the hypersphere, but also serves as the lower bound of the proportion of support vectors, improving parameter selection and model interpretability. Our numerical experiments on two image datasets and five recorded benchmark datasets fully demonstrate the superior performance of CD-SVDD in prediction and computational efficiency, as well as the ν-property. However, the performance of CD-SVDD is still closely tied to the structure of the neural network, and it remains a challenge to find a structure that can consistently perform well on all datasets. One of our future goals is to explore the use of more complex neural network architectures, such as long-short-term memory (LSTM), to enhance the ability of our method to handle sequential data.

REFERENCES

[1] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 1–58, Jul. 2009.

[2] G. Pang, C. Shen, L. Cao, and A. V. D. Hengel, "Deep learning for anomaly detection: A review," *ACM Comput. Surv.*, vol. 54, no. 2, pp. 1–38, Mar. 2021.

[3] M. Jin, A. Lv, Y. Zhu, Z. Wen, Y. Zhong, Z. Zhao, J. Wu, H. Li, H. He, and F. Chen, "An anomaly detection algorithm for microservice architecture based on robust principal component analysis," *IEEE Access*, vol. 8, pp. 226397–226408, 2020.

[4] M. M. Moya, M. W. Koch, and L. D. Hostetler, "One-class classifier networks for target recognition applications," NASA, STI/Recon, Washington, DC, USA, Tech. Rep., 1993, vol. 93, p. 24043.

[5] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Comput.*, vol. 13, no. 7, pp. 1443–1471, Jul. 2001.

[6] S. Levikari, T. J. Kärkkäinen, C. Andersson, J. Tamminen, M. Nykyri, and P. Silventoinen, "Nondestructive acoustic testing of ceramic capacitors using one-class support vector machine with automated hyperparameter selection," *IEEE Access*, vol. 8, pp. 226337–226351, 2020.

[7] D. M. J. Tax and R. P. W. Duin, "Support vector data description," *Mach. Learn.*, vol. 54, no. 1, pp. 45–66, Jan. 2004.

[8] H. Lian, "On feature selection with principal component analysis for one-class SVM," *Pattern Recognit. Lett.*, vol. 33, no. 9, pp. 1027–1031, Jul. 2012.

[9] B. S. Kumar and V. Ravi, "Text document classification with PCA and one-class SVM," in *Proc. 5th Int. Conf. Frontiers Intell. Comput., Theory Appl.*, vol. 1. New York, NY, USA: Springer, 2017, pp. 107–115.

[10] F. Shen, Z. Song, and L. Zhou, "Improved PCA-SVDD based monitoring method for nonlinear process," in *Proc. 25th Chin. Control Decis. Conf. (CCDC)*, Guizhou, China, May 2013, pp. 4330–4336.

[11] F. K. Alarfaj, N. A. Khan, M. Sulaiman, and A. M. Alomair, "Application of a machine learning algorithm for evaluation of stiff fractional modeling of polytropic gas spheres and electric circuits," *Symmetry*, vol. 14, no. 12, p. 2482, Nov. 2022.

[12] H. Alhakami, M. Kamal, M. Sulaiman, W. Alhakami, and A. Baz, "A machine learning strategy for the quantitative analysis of the global warming impact on marine ecosystems," *Symmetry*, vol. 14, no. 10, p. 2023, Sep. 2022.

[13] H. Alhakami, N. A. Khan, M. Sulaiman, W. Alhakami, and A. Baz, "On the computational study of a fully wetted longitudinal porous heat exchanger using a machine learning approach," *Entropy*, vol. 24, no. 9, p. 1280, Sep. 2022.

[14] N. A. Khan, M. Sulaiman, J. Seidu, and F. S. Alshammari, "Investigation of nonlinear vibrational analysis of circular sector oscillator by using cascade learning," *Adv. Mater. Sci. Eng.*, vol. 2022, pp. 1–15, Sep. 2022.

[15] A. Ali, S. Qadri, W. K. Mashwani, W. Kumam, P. Kumam, S. Naeem, A. Goktas, F. Jamal, C. Chesneau, S. Anam, and M. Sulaiman, "Machine learning based automated segmentation and hybrid feature analysis for diabetic retinopathy classification using fundus image," *Entropy*, vol. 22, no. 5, p. 567, May 2020.

[16] A. L. Alfeo, M. G. C. A. Cimino, G. Manco, E. Ritacco, and G. Vaglini, "Using an autoencoder in the design of an anomaly detector for smart manufacturing," *Pattern Recognit. Lett.*, vol. 136, pp. 272–278, Aug. 2020.

[17] Z. Wang and Y.-J. Cha, "Unsupervised deep learning approach using a deep auto-encoder with a one-class support vector machine to detect damage," *Struct. Health Monit.*, vol. 20, no. 1, pp. 406–425, Jan. 2021.

[18] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft, "Deep one-class classification," in *Proc. ICML*, Stockholm. Sweden, 2018, pp. 4393–4402.

[19] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," 2019, *arXiv:1901.03407*.

[20] Y. Lei, T. Hu, and K. Tang, "Generalization performance of multi-pass stochastic gradient descent with convex loss functions," *J. Mach. Learn. Res.*, vol. 22, no. 1, pp. 1145–1185, 2021.

[21] B. Schölkopf, A. J. Smola, R. C. Williamson, and P. L. Bartlett, "New support vector algorithms," *Neural Comput.*, vol. 12, no. 5, pp. 1207–1245, May 2000.

[22] W.-C. Chang, C.-P. Lee, and C.-J. Lin, "A revisit to support vector data description," Dept. Comput. Sci., Nat. Taiwan Univ., Taiwan, Tech. Rep., 2013. [Online]. Available: https://leepei.github.io/paper/svdd/svdd.pdf

[23] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge Univ. Press, 2004.

[24] H.-J. Chiu, T. S. Li, and P.-H. Kuo, "Breast cancer–detection system using PCA, multilayer perceptron, transfer learning, and support vector machine," *IEEE Access*, vol. 8, pp. 204309–204324, 2020.

[25] I. Steinwart and A. Christmann, *Support Vector Machines*. New York, NY, USA: Springer, 2008.

[26] R. Kohavi, "Scaling up the accuracy of Naive–Bayes classifiers: A decision-tree hybrid," in *Proc. KDD*, vol. 96. Portland, OR, USA: AAAI Press, 1996, pp. 202–207.

[27] A. V. Uzilov, J. M. Keegan, and D. H. Mathews, "Detection of non-coding RNAs on the basis of predicted secondary structure formation free energy change," *BMC Bioinf.*, vol. 7, no. 1, pp. 1–30, Dec. 2006.

[28] R. G. Andrzejak, K. Lehnertz, F. Mormann, C. Rieke, P. David, and C. E. Elger, "Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 64, no. 6, Nov. 2001, Art. no. 061907.

[29] N. Roberts, D. R. Lorimer, and M. Kramer, *Handbook of Pulsar Astronomy*, vol. 4. New York, NY, USA: Cambridge Univ. Press, 2005.

[30] D. Prokhorov, T. Feldkamp, L. Feldkamp, and K. Marko, "IJCNN 2001 neural network competition," in *Proc. IJCNN*, 2001, pp. 1–25.

[31] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," CIFAR, Univ. Toronto, Toronto, ON, Canada, Tech. Rep., 2009. [Online]. Available: https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf

[32] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *Proc. 8th IEEE Int. Conf. Data Mining*, Pisa, Italy, Dec. 2008, pp. 413–422.

[33] J. Babaud, A. P. Witkin, M. Baudin, and R. O. Duda, "Uniqueness of the Gaussian kernel for scale-space filtering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 1, pp. 26–33, Jan. 1986.

[34] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. 13th Int. Conf. Artif. Intell. Statist.*, Sardinia, Italy, 2010, pp. 249–256.

[35] C.-J. Hsieh, K.-W. Chang, C.-J. Lin, S. S. Keerthi, and S. Sundararajan, "A dual coordinate descent method for large-scale linear SVM," in *Proc. 25th Int. Conf. Mach. Learn.*, 2008, pp. 408–415.

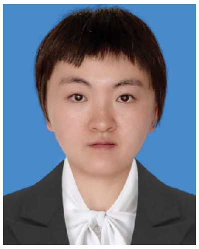
[36] J. Myerson, L. Green, and M. Warusawitharana, "Area under the curve as a measure of discounting," *J. Exp. Anal. Behav.*, vol. 76, no. 2, pp. 235–243, 2001.

[37] W. J. Conover, *Practical Nonparametric Statistics*. New York, NY, USA: Wiley, 1999.

RENXUE JIANG was born in China, in 1997. He received the B.S. degree in economic statistics from Beihua University, Jilin, China. He is currently pursuing the M.S. degree in statistics with the Yunnan University of Finance and Economics, Yunnan, China.

His research interests include support vector machines and deep neural networks in computer vision and natural language processing.

Mr. Jiang received several awards, including the National Encouragement Scholarship and the Yunnan Provincial Government Scholarship.



ZHIJI YANG (Member, IEEE) was born in China, in 1989. She received the Ph.D. degree from the College of Science, China Agricultural University, Beijing, China, in 2018.

She is currently an Associate Professor with the Department of Statistics, Yunnan University of Finance and Economics, Kunming, China. Her research has appeared in IEEE Transactions on Neural Networks and Learning Systems, *Pattern Recognition*, *Knowledge-Based Systems*, *Neurocomputing*, and *Engineering Applications of Artificial Intelligence*. Her current research interests include statistical machine learning, operation research, and data mining.

JIANHUA ZHAO (Senior Member, IEEE) received the Ph.D. degree in statistics from The University of Hong Kong, Hong Kong, in 2009.

He is currently a Full Professor with the Department of Statistics, Yunnan University of Finance and Economics, Kunming, China. He has authored more than 20 peer-reviewed papers in journals and conferences, including IEEE Transactions on Neural Networks and Learning Systems, IEEE Transactions on Neural Networks, and IEEE Transactions on Cybernetics. His current research interests include statistical machine learning, data science, and computational statistics.