Neurocomputing 453 (2021) 131–140

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

VAE-based Deep SVDD for anomaly detection

Yu Zhou *, Xiaomin Liang, Wei Zhang, Linrang Zhang, Xing Song

National Key Lab. of Radar Signal Processing, Xidian Univ., Xi'an 710071, China

ARTICLE INFO

Article history:
Received 3 December 2020
Revised 9 April 2021
Accepted 25 April 2021
Available online 29 April 2021
Communicated by Zidong Wang

2010:
00-01
99-00

Keywords:
Anomaly detection
Variational autoencoder
Deep support vector data description

ABSTRACT

Anomaly detection is an essential task for different fields in the real world. The imbalanced data and lack of labels make the task challenging. Deep learning models based on autoencoder (AE) have been applied to address the above difficulties successfully. However, in these AE-based deep methods, the AE-based model's optimization and the anomaly detector design are separated. Therefore, the latent representations in AE are less relevant for the anomaly detection task, which reduces the accuracy of anomaly detection. A deep support vector data description based on variational autoencoder (Deep SVDD-VAE) is proposed in this paper to solve this problem. In the proposed model, VAE is used to reconstruct the input instances, while a spherical discriminative boundary is learned with the latent representations simultaneously based on SVDD. Unlike existing AE-based methods, we seek the model parameters via the joint optimization of VAE and SVDD, which ensures the separability of the latent representations. Experimental results on MNIST, CIFAR-10, and GTSRB datasets show the effectiveness of Deep SVDD-VAE.
© 2021 Elsevier B.V. All rights reserved.

1. Introduction

Hawkins defined an anomaly as an observation that deviates from the other observations so much as to arouse suspicion that it was generated by a different mechanism [1]. Detecting anomaly is an essential preprocessing step to analyze or remove the erroneous data. Anomaly detection is widely applied in network intrusion detection[2], fraud detection [3], surveillance videos [4,5], medical diagnosis [6–8], and numerous other fields. Anomaly detection can sometimes be treated as a binary classification task. Some machine learning techniques are widely used for anomaly detection tasks, such as One-Class SVM (OCSVM) [9], Kernel Density Estimation [10], and Support Vector Data Description (SVDD) [11].

Recently, deep methods have attracted more attention because they can capture richer representations, among which the techniques based on autoencoder (AE) [12] are popular in anomaly detection. An AE is a feed-forward neural network that tries to reconstruct the input instances at the output layer, which has emerged as an effective method for anomaly detection [13]. In AE-based deep methods [14–16], autoencoder is used to learn a representation model for normal instances by minimizing reconstruction errors. Therefore, abnormal instances cannot be well reconstructed by AE, and the instances with high reconstruction error will be regarded as abnormal. Variational auto-encoder (VAE) [17] based anomaly detection method [18,19] also discriminates the normal and abnormal data based on reconstruction. However, the methods above are designed to imitate original data distribution and only focus on reconstruction errors [20]. These works do not provide a customized classifier for anomaly detection. They usually do not directly contribute to detecting anomalies [21]. The learned features are indiscriminative, which may lead to unstable performance. Autoencoder is combined with traditional technologies to solve this problem, which is called hybrid models. In these hybrid models [22–25], the features are obtained by AE or VAE, and they are utilized by traditional technologies to perform anomaly detection. Yao R et al. [26] trained a VAE to learn the property of data and use the trained model's encoder as the feature extractor. Then Four traditional anomaly detection methods are performed to detect the features represented by the proposed model. Z. Ghrib et al. [27] propose hybrid method composed of an LSTM autoencoder on normal records to learn the representations. Then SVM is used to accurately detect abnormal records. A drawback of these hybrid models is that the extracted features are separated from the anomaly detection and may be suboptimal or less relevant to the anomaly detection task [28]. To overcome the above drawback, in recent works [28,29], it is proposed that the training of model should be driven by anomaly detection based objective function. The anomaly detection based

* Corresponding author.
E-mail address: zhouyu@mail.xidian.edu.cn (Y. Zhou).

https://doi.org/10.1016/j.neucom.2021.04.089
0925-2312/© 2021 Elsevier B.V. All rights reserved.

objective function is modified based on the objective function of classical anomaly detection techniques. Specifically, features are extracted via deep networks with an SVM-like [28] or SVDD-like [29] objective function. Classical anomaly detection methods and deep networks are joined together to supplement mutually. The combination endows these joint models with the advantages of learning rich features and creating a close-fitting boundary around the normal data. They both obtain better performance over the previous methods. However, because the objective function of [28] is non-convex, the solution may not be the global optimum. In [29], a phenomenon named "hypersphere collapse" may occur in the proposed model, which means that the network maps all data points into one point in the latent space. These works inspire us to propose a method, which can not only combine the benefits of classical anomaly detection methods and deep networks but also avoid the above problems. Motivated by the above reasons, we propose a Deep SVDD based on variational autoencoder (Deep SVDD-VAE) to detect the anomaly. In Deep SVDD-VAE, we incorporate the traditional anomaly detection technique, i.e., SVDD, into the feature learning model, i.e., VAE. Compared with hybrid models, the combination of SVDD and VAE in our method is no longer separate. The model is encouraged to condense normal data as close together as possible at a sphere-like region and keep normal points from overlapping in the latent space. Besides, we theoretically show that our work can avoid the collapse of the hypersphere without adding additional restrictions compared with the work in [29]. Our main contributions are summarized as follows:

- Our method is a combination of VAE and SVDD. Unlike hybrid models, VAE and SVDD are not used separately in different stages. Specifically, the model parameters of the VAE model and SVDD are jointly learned with a customized objective function to ensure the relevance of latent representations for the anomaly detection task;
- We theoretically show that the combination of deep networks and SVDD in our method can avoid hypersphere collapseand can get rid of additional restrictions.

The remainder of the paper is organized as follows. We introduce the preliminary of our method in Section 2. Our proposed method is presented in Section 3. Section 4 shows the experimental dataset, compared methods, and the results. Finally, Section 5 concludes the paper.

2. Preliminary

2.1. Variational Autoencoder

Variational Autoencoder (VAE) [17] is a generative model. With $\mathbf{x}$ denoted as input and $\mathbf{z}$ denoted as latent representation, VAE is composed of two parts: 1) an encoder $q_\phi(\mathbf{z}|\mathbf{x})$ as recognition model representing the approximation to the intractable true posterior $p_\theta(\mathbf{z}|\mathbf{x})$ and 2) a decoder $p_\theta(\mathbf{x}|\mathbf{z})$ as generative model to generate new data with latent representation $\mathbf{z}$. The architecture of VAE is illustrated in Fig. 1.

Given a dataset $D_n = \{\mathbf{x}_1, \mathbf{x}_2 \cdots \mathbf{x}_n\}$, the training objective of VAE is to solve the maximum likelihood $\sum_{i=1}^{n} \log p_\theta(\mathbf{x}_i)$ with respect to the parameters $\boldsymbol{\phi}$ of encoder and the parameters $\boldsymbol{\theta}$ of decoder, where $\log p_\theta(\mathbf{x}_i)$ can be expressed as:

$$\log\ p_\theta(\mathbf{x}_i) = D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}_i)||p_\theta(\mathbf{z})) + L(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}_i) \tag{1}$$

where $p_\theta(\mathbf{z})$ is the prior over latent variables and $D_{KL}(\cdot)$ denotes Kullback–Leibler (KL) divergence. The second term in the right hand $L(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}_i)$ is called the evidence variational lower bound (ELBO) on the marginal likelihood of datapoint $\mathbf{x}_i$. Since the first term KL-divergence is non-negative, (1) can be written as:

$$\log p_\theta(\mathbf{x}_i) \geqslant L(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}_i) \tag{2}$$

Because $\log p_\theta(\mathbf{x}_i)$ is intractable, ELBO is maximized instead to estimate the maximum likelihood $\log p_\theta(\mathbf{x}_i)$. The ELBO can be written as follows:

$$L(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}_i) = -D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}_i)||p_\theta(\mathbf{z})) + \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x}_i)}[\log p_\theta(\mathbf{x}_i|\mathbf{z})] \tag{3}$$

The second term in (3) is regarded as an expected negative reconstruction error between input data and output data, which requires a random latent variable $\mathbf{z}$ sampling from the approximate posterior $p_\theta(\mathbf{x}_i|\mathbf{z})$. However, back-propagation cannot deal with random variable $\mathbf{z}$. When $q_\phi(\mathbf{z}|\mathbf{x}) \sim \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}, \boldsymbol{\sigma}^2)$, each random variable $\mathbf{z_i}$ can be reparametrized as a differentiable transformation of a noise variable $\boldsymbol{\epsilon}_l \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$: $\mathbf{z}_i^l = \boldsymbol{\mu}_i + \boldsymbol{\sigma}_i \odot \boldsymbol{\epsilon}_l$. (3) can be rewritten as:

$$L(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}_i) = -D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}_i)||p_\theta(\mathbf{z})) + \frac{1}{L}\log p_\theta(\mathbf{x}_i|\mathbf{z}_i^l) \tag{4}$$

The prior distribution $p_\theta(\mathbf{z})$ is assumed to be the isotropic unit Gaussian $\mathcal{N}(\mathbf{0}, \mathbf{I})$, where $\mathbf{I}$ is the identity matrix. The maximum of first term $-D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}_i)||p_\theta(\mathbf{z}))$ means that $q_\phi(\mathbf{z}|\mathbf{x})$ and $p_\theta(\mathbf{z})$ obey the same distribution. To optimize the KL-divergence between $q_\phi(\mathbf{z}|\mathbf{x})$ and $p_\theta(\mathbf{z})$ under the assumption that $p_\theta(\mathbf{z})$ follows Gaussian distribution, the encoder estimates the parameter vectors of the Gaussian distribution $q_\phi(\mathbf{z}|\mathbf{x})$: mean $\boldsymbol{\mu}$ and standard deviation $\boldsymbol{\sigma}$, i.e. $q_\phi(\mathbf{z}|\mathbf{x}) \sim \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}, \boldsymbol{\sigma}^2)$. The explicit expression for KL-divergence between $q_\phi(\mathbf{z}|\mathbf{x})$ and $p_\theta(\mathbf{z})$ can be simply written as:

$$\begin{aligned} &-D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}_i)||p_\theta(\mathbf{z})) \\ &= -D_{KL}\left(\mathcal{N}\left(\boldsymbol{\mu}_i, \boldsymbol{\sigma}^{i^2}\right)\right)||\mathcal{N}(\mathbf{0}, \mathbf{I}) \\ &= \frac{1}{2}\sum_{j=1}^{J}\left(1 + log\left(\left(\boldsymbol{\sigma}_i^j\right)^2\right) - \left(\boldsymbol{\mu}_i^j\right)^2 - \left(\boldsymbol{\sigma}_i^j\right)^2\right) \end{aligned} \tag{5}$$

where J is the dimension of z, $\boldsymbol{\mu}_i^j$ and $\boldsymbol{\sigma}_i^j$ denote the j^{th} element of $\boldsymbol{\mu}_i$ and $\boldsymbol{\sigma}_i$ respectively. The objective function of VAE at $\mathbf{x}_i$ is as follows:

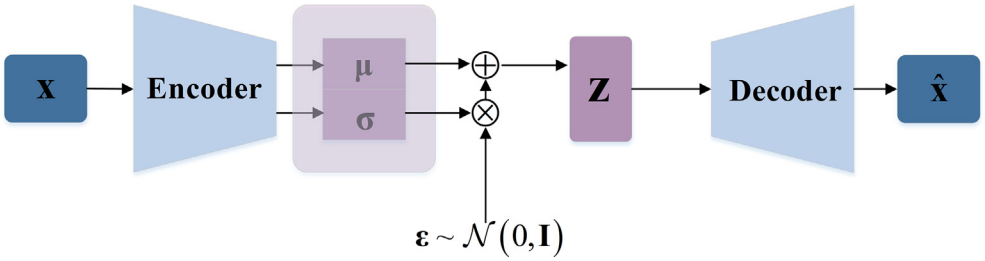


Fig. 1. The architecture of VAE.

$$L(\boldsymbol{\theta},\boldsymbol{\phi};\mathbf{x}_i)=\frac{1}{2}\sum_{j=1}^{J}\left(1+log\left(\left(\boldsymbol{\sigma}_i^j\right)^2\right)-\left(\boldsymbol{\mu}_i^j\right)^2-\left(\boldsymbol{\sigma}_i^j\right)^2\right)+\frac{1}{L}\times\log p_{\boldsymbol{\theta}}(\mathbf{x}_i|\mathbf{z}_i^l) \tag{6}$$

As the objective function (6) is optimized by stochastic gradient ascent, VAEs learn the recognition model parameters $\boldsymbol{\phi}$ jointly with the generative model parameter $\boldsymbol{\theta}$. Given an input $\mathbf{x}_i$, the encoder of VAE generates parameters of the approximate posterior at this data point $\boldsymbol{\mu}_i$ and $\boldsymbol{\sigma}_i$. The latent representation obtained through $\mathbf{z}_i^l=\boldsymbol{\mu}_i+\boldsymbol{\sigma}_i\odot\boldsymbol{\epsilon}_l$, where $\boldsymbol{\varepsilon}_l\sim\mathcal{N}(\mathbf{0},\mathbf{I})$, is transferred to decoder to generate reconstruction $\hat{\mathbf{x}}_i$.

2.2. Support Vector Data Description

Support Vector Data Description (SVDD) [11] is a one-class classification technique, which uses a learned hypersphere to separate the data. All samples in the input space $\mathcal{X}\subseteq\mathbb{R}^d$ are then mapped into the feature space $\mathcal{F}_k$ via a mapping function $\varphi(\cdot)$. The purpose of $\varphi(\cdot)$ is to make the decision boundary more compact in the feature space than in the input space. Normal samples will fall into the boundary while other samples will fall outside the boundary in the feature space. For SVDD, the boundary is called hypersphere. The objective of SVDD is to find the smallest hypersphere with center $\mathbf{c}\in\mathcal{F}_k$ and radius R that can enclose the majority of the normal data in the feature space. Given a dataset $D_n=\{\mathbf{x}_1,\mathbf{x}_2\cdots\mathbf{x}_n\}$ with $\mathbf{x}_i\in\mathcal{X}$, the primal problem of SVDD is given by

$$\begin{aligned}&\min_{R,c,\xi}R^2+\gamma\sum_{i=1}^{n}\xi_i\\ s.t.\ &\|\varphi(\mathbf{x}_i-\mathbf{c})\|\leqslant R^2+\xi_i,\ i=1,2\cdots n,\\ &\xi_i\geqslant 0,\ i=1,2\cdots n.\end{aligned} \tag{7}$$

where $\|\cdot\|$ represents the Euclidean norm and $\boldsymbol{\xi}=(\xi_1,\xi_2\cdots\xi_n)^T$ are slack variables which can penalty violations to relax the constraints and make the decision boundary soft. γ is a parameter that controls the trade-off between the sphere volume and the sum of penalties $\sum_{i=1}^{n}\xi_i$. By introducing Lagrange function, the dual problem can be written as follows:

$$\begin{aligned}&\max_{\alpha}\sum_{i=1}^{n}\alpha_i\langle\varphi(\mathbf{x}_i),\varphi(\mathbf{x}_i)\rangle-\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j\langle\varphi(\mathbf{x}_i),\varphi(\mathbf{x}_j)\rangle\\ &s.t.\ 0\leqslant\alpha_i\leqslant\gamma\\ &\sum_{i=1}^{n}\alpha_i=1,\ i=1,2\cdots n.\end{aligned} \tag{8}$$

where α_i for $i=1,2\cdots n$ is the Lagrange multipliers. In fact, we do not depend on $\varphi(\cdot)$ to get the inner products $\varphi(\mathbf{x}_i)\cdot\varphi(\mathbf{x}_j)$ directly because there is no explicit expression for $\varphi(\mathbf{x}_i)$. Instead, kernel function $K(\cdot)$ is employed to get the inner products that equals to $K(\mathbf{x}_i,\mathbf{x}_j)$. Therefore, the dual problem is changed into:

$$\begin{aligned}&\max_{\alpha}\sum_{i=1}^{n}\alpha_iK(\mathbf{x}_i,\mathbf{x}_i)-\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_jK(\mathbf{x}_i,\mathbf{x}_i)\\ &s.t.\ 0\leqslant\alpha_i\leqslant\gamma\\ &\sum_{i=1}^{n}\alpha_i=1,\ i=1,2\cdots n.\end{aligned} \tag{9}$$

Problem (9) is a standard quadratic optimization problem. From the Kuhn-Tucker condition, the center $\mathbf{c}$ can be expressed as $\mathbf{c}=\sum_{i=1}^{n}\alpha_i\varphi(\mathbf{x}_i)$. The samples $\mathbf{x}_i$ with $\alpha_i\neq 0$ are called Support Vectors (SVs). Assume $\mathbf{x}_k$ is one of the SVs and $0\leqslant\alpha_i\leqslant\gamma$holds true, R can be calculated as follows:

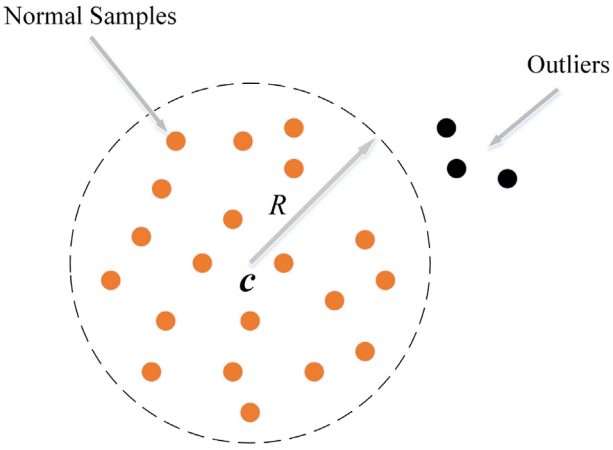


Fig. 2. SVDD for anomaly detection.

$$R^2=K(\mathbf{x}_k,\mathbf{x}_k)-2\sum_{i=1}^{n}\alpha_iK(\mathbf{x}_i,\mathbf{x}_k)+\sum_{i=1}^{n}\sum_{i=1}^{n}\alpha_i\alpha_jK(\mathbf{x}_i,\mathbf{x}_j) \tag{10}$$

For a test sample $\mathbf{X}_{test}$, it is deemed as anomalous if the distance between it and the sphere center $\mathbf{c}$ is greater than the radius R as illustrated in Fig. 2.

3. Proposed method

3.1. Overview

In this paper, we propose a deep SVDD based on VAE for the anomaly detection task. The proposed model consists of a VAE as a feature extractor and an SVDD as a one-class classifier. The flowchart of Deep SVDD-VAE is shown in Fig. 3. Deep SVDD-VAE firstly obtains the features via the encoder of VAE. The extracted features are transferred to subsequent SVDD for learning a hypersphere while the decoder of VAE utilizes these features to reconstruct input instances. Deep SVDD-VAE jointly optimizes these two processes. In the evaluation stage, the instances falling out the learned hypersphere are regarded as anomalous.

3.2. The objective function of SVDD in the model

As is introduced in Section 3.1, SVDD aims to construct a hypersphere separating the normal and abnormal patterns. Under the assumption that all instances in the training set are normal, modified SVDD is adopted to build the Deep SVDD-VAE model. In the modified SVDD, the original problem can be simplified as follows:

$$\min_{\mathbf{c},\mathcal{W}}\frac{1}{n}\sum_{i=1}^{n}\|\varphi(\mathbf{x}_i;\mathcal{W}-\mathbf{c})\|^2 \tag{11}$$

where $\varphi(\mathbf{x}_i;\mathcal{W})$ is the latent representation of the datapoint $\mathbf{x}_i$, and $\mathbf{c}$ is a parameter learned by the network. In modified SVDD, the mean distance between all representations and the learned center $\mathbf{c}$ is punished, not the radius R. The main difference between original and modified SVDD is that slack variables are no longer penalized because it is assumed that the training set does not contain any abnormal data. The network $\varphi(\cdot)$ is required to map all instances close to the center $\mathbf{c}$. Ruff L et al. [29] theoretically demonstrated that improperly formulated $\varphi(\cdot)$ or hypersphere center $\mathbf{c}$ can make deep SVDD learn trivial, uninformative solutions. Therefore, three network restrictions about center, bias terms, and activation functions are proposed in [29]. To summarize briefly, the center cannot be a free variable and bias terms or bounded activation functions should not be used in deep SVDD. These restrictions are proposed

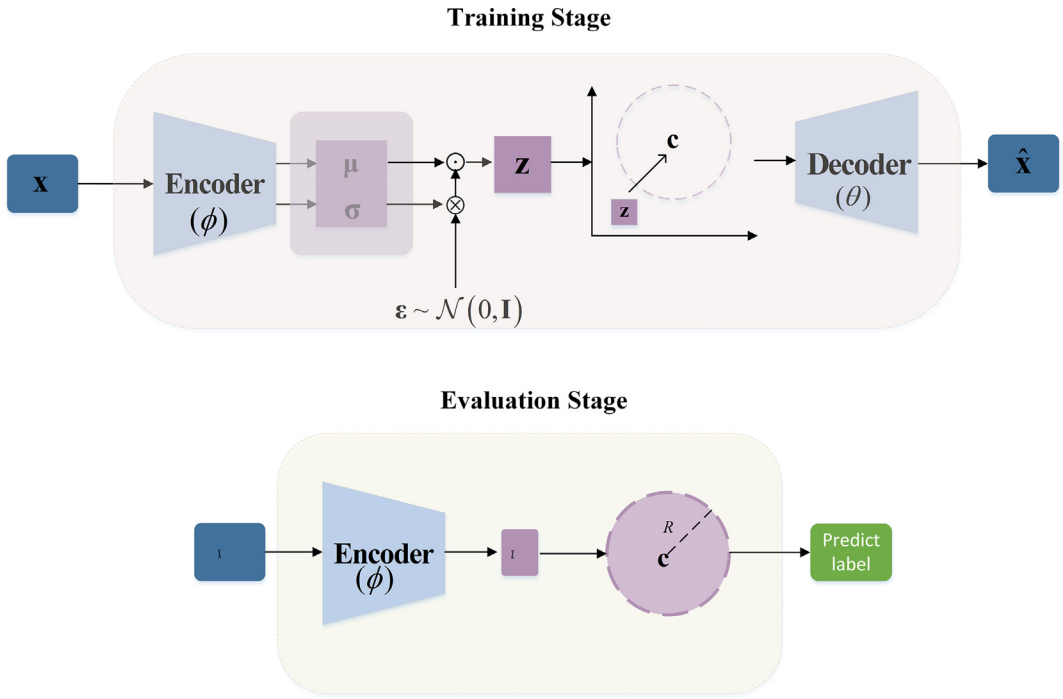


Fig. 3. The flowchart of Deep SVDD-VAE.

for avoiding a phenomenon named hypersphere collapse occurring in their model. The phenomenon refers to the situation where all representations coincide in the latent space. In this situation, the nearly-zero radius cannot support the hypersphere. By specifying VAE as the feature extractor $\varphi(\cdot)$, our work can avoid hypersphere collapse in theory. More details will be explained in Section 3.3.

3.3. Deep SVDD-VAE model

VAE is adopted to extract features and reconstruct inputs to avoid learned features being completely similar. If all input instances were mapped into one point in the latent space, the extracted features owned by all input instances would be less helpful to reconstruct different input instances and lead to a high reconstruction loss in the VAE-based model. Moreover, the representations transferred by VAE involves randomly sampled noise, making it nearly impossible for all representations to coincide.

The aim of our proposed method is to minimize the average distance between all representations and center $\mathbf{c}$ while keeping the ability to reconstruct input data. Given a training dataset $D_n = \{\mathbf{x}_1, \mathbf{x}_2 \cdots \mathbf{x}_n\}$, we define the Deep SVDD-VAE objective function as:

$$\min_{\mathbf{c},w} \frac{\alpha}{n}\sum_{i=1}^{n}\left[D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}_i)||p_\theta(\mathbf{z})) - \frac{1}{L}\sum_{i=1}^{L}\log p_\theta(\mathbf{x}_i|\mathbf{z}_i^l)\right] + \frac{1}{n}\sum_{i=1}^{n}\|q_\phi(\mathbf{z}|\mathbf{x}_i) - \mathbf{c}\| \tag{12}$$

where ϕ is the parameter of the encoder of VAE, and θ is the parameter of the decoder of VAE. α is a tunable parameter and is fixed in the training stage, which can balance the impacts of VAE and SVDD to the whole model.

In theory, the optimal solution is required to map all input into different outputs in the latent space. Different outputs can support the hypersphere not to collapse. Three restrictions that are proposed to avoid hypersphere collapse can be removed. The center can be learned by the network rather than be set in advance like [29]. Under the circumstance that most of the training data is normal, it can be inferred that center $\mathbf{c}$ is more likely to be the center of all latent representations if $\mathbf{c}$ is a free variable. In the evaluation stage, it is assumed that the representation of the normal data is closer to the center than that of anonymous data in the latent space.

Compared with the original VAE, the main difference between VAE and Deep SVDD-VAE is whether to restrict the distribution of latent variables. Unlike previous methods based on autoencoders, the model is encouraged to extract more common features for anomaly detection. To minimize Deep SVDD-VAE loss, the model is required to find a solution mapping the latent representation to the center as close as possible and reconstructing input data as well as possible. Moreover, as a generative model, instead of directly modeling the latent variable, VAE is designed to learn its distribution parameters. Although normal instances and anomaly have similar features, the representations of them can be different because the variability of the latent space is taken into account. The characteristic endows our method with greater expressive power.

3.4. Anomaly evaluation with Deep SVDD-VAE

According to the type of outputs, anomaly detection techniques can be divided into scoring-based methods and label-based methods. Deep SVDD-VAE is a scoring-based technique. Anomaly score is defined as the distance between representation and center of the trained hypersphere. For a given test point $\mathbf{x}_i$, the anomaly score of $\mathbf{x}_i$ is as follows:

$$s(\mathbf{x}_i) = \|\mathbf{z}_i - \mathbf{c}^*\| \tag{13}$$

where $\mathbf{z}_i$ is the learned representation of $\mathbf{x}_i$ and $\mathbf{c}^*$ is the center of a trained hypersphere. $s(\mathbf{x}_i)$ is used to evaluate the state of $\mathbf{x}_i$. The higher the value $s(\mathbf{x}_i)$, the point $\mathbf{x}_i$ is more likely to be anomalous. The radius R^* is the 95th percentile of anomaly scores of all training instances. The aim of our proposed method is to find a data-enclosing hypersphere with minimum volume in latent space to

describe all training data. In the proposed method, it is assumed that all training data is normal. Therefore, theoretically, the radius can be defined as the largest distance between the feature of training data and the hypersphere center. However, there are also some normal samples deviating from most normal data in the training set. The choice of largest distance for R^* may be inappropriate and result in poor performance on anomaly detection. We think that the situation should be taken into account and a trade-off between the accuracy and radius should be made. So, we define radius as the 95th percentile of anomaly scores (distance between the feature of data and center) of all training instances. The anomaly score $s(\mathbf{x}_i)$ will be compared with radius R^* to determine whether the instance is abnormal or not. If the anomaly score is larger than the radius, the instance is regarded as anomalous. The working procedure of Deep SVDD-VAE is constructed as Algorithm 1.

Algorithm 1. The working procedure of Deep SVDD-VAE model

Input: Dataset $D_n = \{\mathbf{x}_1, \mathbf{x}_2 \cdots \mathbf{x}_n\}$,training epoch k;
Initialize model parameters $\boldsymbol{\phi}$,$\boldsymbol{\theta}$, and center $\boldsymbol{c}$
1: **while** $i \leqslant k$ **do**
2: $\mathbf{x} \Leftarrow$ Randomly select mini-batch from training dataset D_n;
3: $\{\boldsymbol{\mu}, \boldsymbol{\sigma}\} \Leftarrow Encoder_{\boldsymbol{\phi}}(\mathbf{x})$;
4: $\mathbf{z} \Leftarrow$ Samples from Normal distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$;
5: $\mathbf{c}_{i+1} \Leftarrow \frac{1}{n}\sum_{i=1}^{n}\mathbf{z}_i$
6: $\hat{\mathbf{x}} \Leftarrow Decoder_{\boldsymbol{\theta}}(\mathbf{z})$;
7: Calculating L as equation 12;
8: $\{\boldsymbol{\phi}_{i+1}, \boldsymbol{\theta}_{i+1}\} \Leftarrow \{\boldsymbol{\phi}_i, \boldsymbol{\theta}_i\} - \nabla_{\boldsymbol{\phi}_i, \boldsymbol{\theta}_i} L$;
9: **end while**
10: **Return:** Model parameters $\phi^* \Leftarrow \phi_k$, $\theta^* \Leftarrow \theta_k$, $\mathbf{c}^* \Leftarrow \mathbf{c}_k$
11: Obtaining the anomaly score of each point $s(\mathbf{x}_i) = \|\mathbf{z}_i - \mathbf{c}^*\|$
12: $R^* \Leftarrow$ 95th percentile of the sorted anomaly scores of all training instances
13: Extracting the feature of each test instance $\mathbf{z}_{test} \Leftarrow Encoder_{\phi^*}(\mathbf{x}_{test})$
14: Calculating the anomaly score of each test instance $s(\mathbf{x}_{test}) = \|\mathbf{z}_{test} - \mathbf{c}^*\|$
15: Comparing $s(\mathbf{x}_{test})$ with R^*: if $s(\mathbf{x}_{test}) > R^*$, $\mathbf{x}_{test}$ is anomaly.

4. Experiments

In this section, different datasets and methods are used to verify the effectiveness of the proposed method. We implement our architecture in Python 3.6 using Keras, with Tensorflow as a backend engine. All the experiments were run on GPU Nvidia GeForce RTX 2080 SUPER.

4.1. Dataset and evaluate criteria

To visually check the anomalies detected by our method, we choose image datasets, including MNIST, CIFAR-10, and GTSRB. Details of the used datasets in our experiments are presented in Table 1.

MNIST [30] &**CIFAR-10** [31]: MNIST consists of 70000 28*28 grayscale images of handwritten digits 0 through 9, and CIFAR-10 is composed of 60000 32*32 color images of real-world objects in 10 classes. Ten anomaly detection tasks are respectively created for both datasets. One of the classes is specified as the normal class, and the samples from other classes are considered anomalies. We only train with samples from normal class in each task, giving a training set of about 6000 samples for MNIST and about 5000 samples for CIFAR-10. Original test sets are transformed into binary datasets with an anomaly ratio of 90% and used to evaluate different methods' performance.

GTSRB [32]: GTSRB contains 43 kinds of traffic signals with the image size varying from 15*15 to 250*250. We only use the stop signs and their adversarial samples crafted by attacking model via FGSM. The training set is composed of 780 stop signs. The test set contains 270 stop signs and 20 adversarial samples.

Evaluation criteria: The datasets used in our experiments are divided into two classes. The samples from specified class (normal class) are labeled with zero and the samples from other classes (abnormal classes) are labeled with 1. Area under curve (AUC) is adopted to evaluate the performance of our model and competing methods. The method with larger AUC performs better on anomaly detection task.

4.2. Baseline methods

In our experiments, the proposed Deep SVDD-VAE method is compared with several methods.

One Class Support Vector Machine (OC-SVM) [9] uses the hyperplane to separate the normal and abnormal data points. Anomaly score is defined as the distance between an instance and the learned hyperplane. Gaussian kernel function is used in our experiment.

Kernel Density Estimation (KDE) [10] requires the kernel function to estimate the actual density for the given data set. Anomaly score is defined as the probability of each instance. The bigger probability means that the instance is more likely to be normal. Gaussian kernel function is selected in this experiment.

Isolation Forest (IF) [33] builds an ensemble of iTrees for a given dataset, called isolation forest. Then instances are passed through all iTrees. The anomaly score is the average of all path lengths. The parameter settings are the same as the original work [33].

One Class Neural Network (OC-NN) [28] trains a neural network to obtain a decision boundary around normal examples. Anomaly score is defined as the distance between the output of each instance and the learned bias. We adopt the same architecture as the encoder of the DCAE for OC-NN, and then apply a feed-forward neural network. The value of parameter, which is equivalent to the percentage of anomalies for each dataset, is set according to respective outlier proportions.

Deep Support Data Vector Description (Deep SVDD) [29] trains a neural network to obtain a hypersphere around all normal data points. We choose One-Class Deep SVDD to compare with our methods under the assumption that there is no anomaly in the

Table 1
Details of the datasets used for performance evaluation.

Dataset	Normal class	Outliers classes	Dimensions	Testing anomaly ratio
MNIST	Single class	Other classes	784	≈ 90%
CIFAR-10	Single class	Other classes	3072	≈ 90%
GTSRB	Single class	Other classes	3072	7%

training set. Anomaly score is defined as the distance between the center and the latent representation of the test instance. The weights of trained CAE are used for initialization.

Convolutional AutoEncoder (CAE) [34] is devoted to reconstructing the input from the latent feature. The reconstruction error of an instance is regarded as the anomaly score.

For the first three methods mentioned above, we reduce the dimensionality of all data via PCA. For the last three methods, CAE, OC-NN and One-Class Deep SVDD share similar architecture of feature extractor with our methods. The details of encoder for Deep SVDD-VAE are listed in the Table 2.

For CAE and Deep SVDD-VAE, the architectures of decoders mirror the architectures of encoders, where max-pooling is substituted with upsampling. Rmsprop is adopted with the learning rate of 0.001 to optimize our algorithm for all datasets. We did not implement experiments on MNIST and CIFAR-10 except OC-NN. The relative reported numbers are taken from [29].

4.3. Results and discussions

4.3.1. Performance on MNIST and CIFAR-10

The results are presented in Table 3. On MNIST, Deep SVDD-VAE outperforms all baseline methods, especially on task '5'. The results are analyzed as follows: compared with reconstruction-based methods like CAE, the model can condense normal samples' features as closely as possible. The common factors of normal instances are fully learned. In the test stage, the instances with more common factors are possible to be deemed as normal. Compared with One-Class Deep SVDD, Deep SVDD-VAE benefits from the expressive power of the VAE and removing the restrictions on the network for avoiding the "hypersphere collapse".

Table 2
Details of the architectures used in our experiments.

Dataset	Architecture	Batch size
MNIST	8×(5×5×1)-filters + BN + leaky Relu 4×(5×5×1)-filters + BN + leaky Relu Dense layer of 32 units	200
CIFAR-10	32×(5×5×3)-filters + BN + leaky Relu 64×(5×5×3)-filters + BN + leaky Relu 128×(5×5×3)-filters + BN + leaky Relu Dense layer of 128 units	200
GTSRB	16×(5×5×3)-filters + BN + leaky Relu 32×(5×5×3)-filters + BN + leaky Relu 64×(5×5×3)-filters + BN + leaky Relu Dense layer of 32 units	64

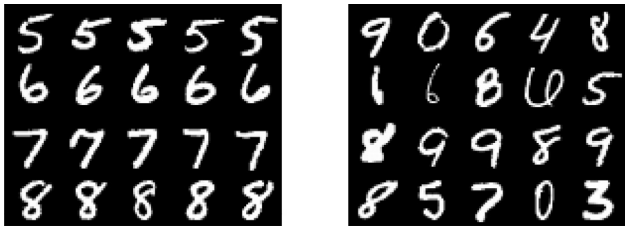


Fig. 4. Samples detected by Deep SVDD-VAE on different tasks. The normal classes from top to bottom are digit 5, digit 6, digit 7 and digit 8. Left: Most normal samples; Right: Misclassified samples.

As we can see in Fig. 4, the samples misclassified by Deep SVDD-VAE (right) have specific similar structures to that of normal samples. These misclassified samples indicate the locations of common features extracted by our model. The global shapes of these misclassified samples are entirely different from that of normal samples. Because the network is forced to obtain common features and reconstruct inputs, the learned features must be analogous but not entirely similar. However, all normal instances usually share some analogous local structures rather than analogous global structures. For those instances with similar global shapes, the global shape may be considered a redundant feature in the training stage. It is hard for the global shape to be a strong common feature learned by our model for both reasons on MNIST.

From Table 4, it can be found that Deep SVDD-VAE shows an overall strong performance compared with One-Class Deep SVDD on CIFAR-10. It is worth noting that KDE and OC-SVM perform better than deep methods in some classes, such as DEER, FROG, and TRUCK. We present the most normal samples detected by Deep SVDD-VAE in Fig. 5 in the test stage. It is found that DEER, FROG (row 1 and row 2), and BIRD have a similar distribution of colors. A large background area covers the real principal parts of normal instances. If the images were correctly cropped, the performances of our method and One-Class Deep SVDD would be better. As for TRUCK (row 3), we think that our method depends on the boundary between sky and road to detect anomalies on this task. The model is confused by the boundary between the sky and sea in the testing stage. The phenomenon inspires us to think about the applicability of our method. It is more suitable for the task where the normal instances' principal features are more salient than irrelevant features.

The training time and testing time of all used methods in the experiment are drawn in Fig. 6.

Because the time cost by different methods are quite different, we take the logarithm of the results to better show all results in the same graph. As we can see from the figures, the training time of OC-NN and One-Class SVDD are longer than CAE. The reason is that OC-NN and One-Class SVDD need to pre-train CAE and then use the

Table 3
The AUC in % of ours and state-of-the-arts on MNIST. The reported performance is averaged over ten seeds.

Normal class	OC-SVM	KDE	IF	CAE	OC-NN	One-Class Deep SVDD	Deep VAE-SVDD
0	98.6	97.1	98.0	97.6	97.0	98.0	**98.8**
1	99.5	98.9	97.3	98.3	99.5	**99.7**	99.5
2	82.5	79.0	88.6	85.4	85.2	91.7	**93.7**
3	88.1	86.2	89.9	86.7	87.1	91.9	**93.7**
4	94.9	87.9	92.7	86.5	93.5	94.9	**95.8**
5	77.1	73.8	85.5	78.2	86.3	88.5	**92.7**
6	96.5	87.6	95.6	94.6	97.3	98.3	**98.6**
7	93.7	91.4	92.0	92.3	92.9	94.6	**95.0**
8	88.9	79.2	89.9	86.5	89.5	93.9	**94.7**
9	93.1	88.2	93.5	90.4	94.6	96.5	**96.6**

Fig. 5. Most normal samples selected by Deep SVDD-VAE for different tasks on CIFAR-10. The normal classes from top to bottom are DEER, FROG and TRUCK.

weights of CAE for model training. The method proposed in this paper does not require pre-training, so the training time is relatively short. Generally speaking, among the deep methods used in the experiments, Deep SVDD-VAE consumes the least time in the training phase. Once all deep models are trained, they consume almost the same amount of time in the testing phase.

4.3.2. Performance on GTSRB stop signs with adversarial attacks

Table 5 reports the relative results about performance on GTSRB of all methods. Generally, the deep methods perform better, and Deep SVDD-VAE performs best. Fig. 7 shows the most anomalous and normal samples detected by Deep SVDD-VAE. On right of Fig. 5, the twenty most anomalous examples are listed. Some of the normal examples are incorrectly detected because improper cutting makes them look strange. They are labeled with normal class, and it is hard to regard them as normal. Other anomalous examples that look like normal stop signs are crafted by attacking normal examples via FGSM. From Table 5 and Fig. 5, it can be inferred that when faced with adversarial attacks, Deep SVDD-VAE may have good robustness. According to [35], higher adversarial robustness means that the features extracted by Deep SVDD-VAE can align nicely with the human eye. The learned features have better interpretability than that of other methods and can help us have a deeper insight into the nature of the network.

Table 5
The AUCs in % of ours and state-of-the-arts on GTSRB stop signs with adversarial attacks. The reported performance is averaged over ten seeds.

Method	AUC
OC-SVM	55.18
KDE	55.73
IF	67.12
CAE	71.26
OC-NN	70.62
One-Class Deep SVDD	82.35
Deep SVDD-VAE	**83.54**

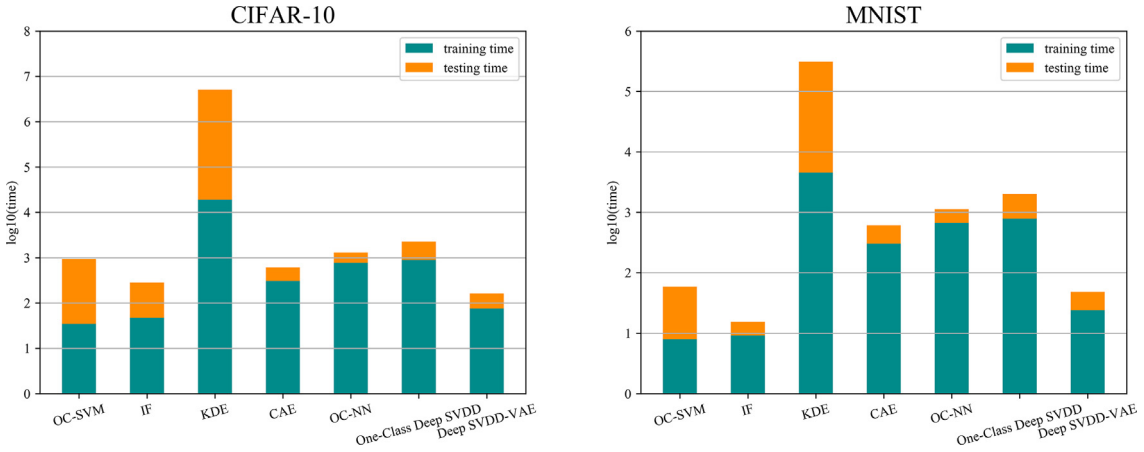


Fig. 6. The training and testing time of different methods. Left: CIFAR-10; Right: MNIST.

Table 4
The AUCs in % of ours and state-of-the-arts on CIFAR-10. The reported performance is averaged over ten seeds.

Normal class	OC-SVM	KDE	IF	CAE	OC-NN	One-Class Deep SVDD	Deep VAE-SVDD
AIRPLANE	61.6	61.2	60.1	59.1	56.8	61.7	**64.4**
AUTOMOBILE	63.8	64.0	50.8	57.4	48.8	**65.9**	65.3
BIRD	50.0	50.1	49.2	48.9	57.4	50.8	**57.5**
CAT	55.9	56.4	55.1	58.4	52.2	59.1	**60.3**
DEER	66.0	**66.2**	49.8	54.0	65.0	60.9	61.6
DOG	62.4	62.4	58.5	62.2	47.6	**65.7**	64.3
FROG	74.7	**74.9**	42.9	51.2	61.9	67.7	66.3
HORSE	62.6	62.6	55.1	58.6	48.5	**67.3**	64.0
SHIP	74.9	75.1	74.2	**76.8**	58.2	75.9	76.5
TRUCK	75.9	**76.0**	58.9	67.3	55.8	73.1	74.8

Fig. 7. Left: The top twenty most anomalous samples selected by Deep SVDD-VAE model. The samples similar to normal samples are crafted adversarial examples; Right: The top twenty most normal samples selected by Deep SVDD-VAE.

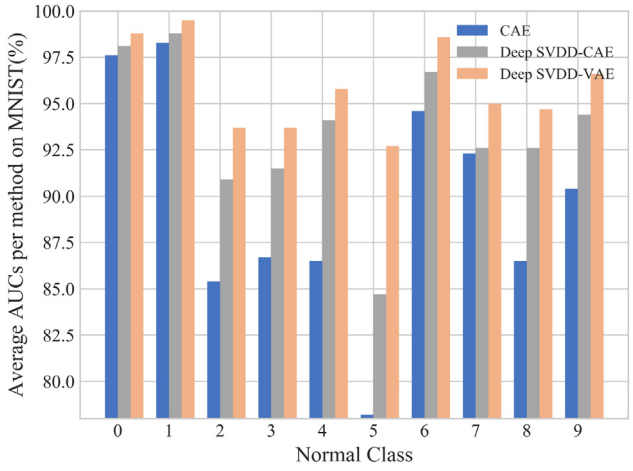


Fig. 8. The AUCs of different methods. Blue bars denote the performance of CAE on different tasks. Grey bars denote the performance of CAE-SVDD while orange bars denote the performance of Deep SVDD-VAE.

4.3.3. Ablation study

In this section, we conduct experiments to illustrate the impact of different components of the model by using CAE instead of using VAE.

We replace the VAE in the proposed model with CAE and carry out the same anomaly detection tasks with Deep SVDD-CAE on MNIST. The corresponding AUCs on MNIST test set are reported in Fig. 8. We can observe that the combination of CAE and SVDD (marked in grey) can help CAE perform better than the original CAE (marked in blue) on anomaly detection tasks. The result indicates that introducing a classical anomaly detection like SVDD into the network's training objective can make full use of the network's ability. As expected, the performance is the best when VAE is used to extract features (marked in orange). The observation above is due to the characteristic of VAEs, as we have introduced in Section 2.1. It is the great expressive power of VAEs that makes the combination more valuable.

4.3.4. Robustness experiments

To verify the robustness of our model, we fix other parameters and change the parameter α. The results are drawn in Fig. 9. Six tasks on MINST and CIFAR-10 are randomly selected, including task '0', task'5', task '8', task 'airplane', task 'bird', and task 'cat'. We unevenly choose α to study its impact to the whole model. The results are consistent with our conclusion in Section 4.3.1. The choice of α is related to the characteristics of the normal class. The parameter can be 1 or larger than 1 on task for normal samples with salient features like task '0','5','8','cat'. On tasks for normal samples with large number of irrelevant features, the parameter α should be smaller than 1 because larger α forces our model to learn more features unrelated to the real class. In general, the choice of parameter α has a limited influence on the performance with 1–2 % float.

5. Conclusion

In this paper, we present a VAE-based anomaly detection method Deep SVDD-VAE by training a VAE on an anomaly detection based objective. Deep SVDD-VAE jointly trains a VAE while optimizing a data-enclosing hypersphere in latent space. With the help of a customized detection objective, VAE is encouraged to condense all features in the latent space and construct inputs as well as possible, giving a full description of normal samples. Moreover, we theoretically prove our method will not lead to "hypersphere collapse". Different experiments are performed to demonstrate that our method is suitable for anomaly detection tasks and exhibits comparable performance over state-of-the-arts. Finally, ablation analyses show that the importance of our method's different components and indicate the possibility of combining SVDD with other autoencoders for a more excellent performance.

CRediT authorship contribution statement

Yu Zhou: Writing - original draft. **Xiaomin Liang:** Conducting experiments and editing. **Wei Zhang:** Methodology, Writing -

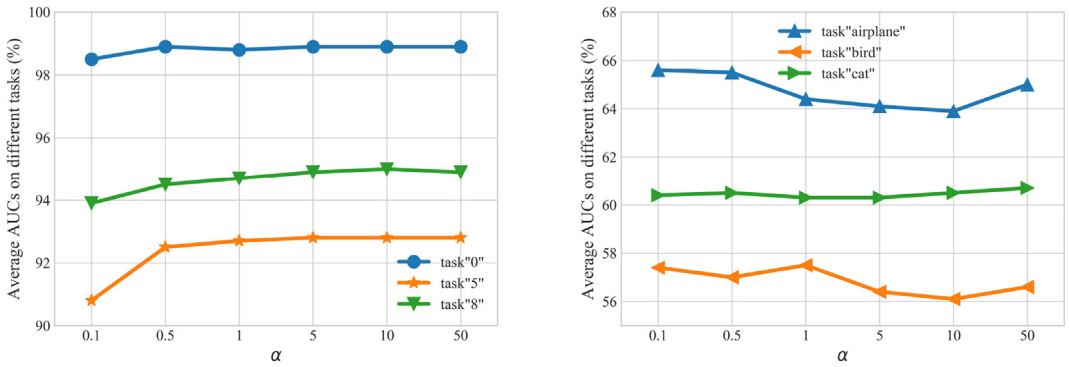


Fig. 9. AUC curves of different tasks with different α. Left: MNIST; Right: CIFAR-10.

review & editing. **Linrang Zhang:** Supervision. **Xing Song:** Investigation.

Declaration of Competing Interest

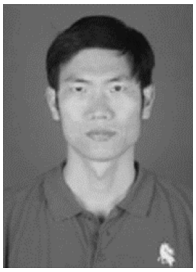
The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work was supported by National Natural Science Foundation of China (Grant Nos. 61871305).

References

[1] D.M. Hawkins, Identification of outliers, vol. 11, Springer, 1980. doi:10.1007/978-94-015-3994-4..
[2] H. Neuschmied, M. Winter, K. Hofer-Schmitz, B. Stojanovic, U. Kleb, Two stage anomaly detection for network intrusion detection, in: Proceedings of the ICISSP..
[3] T. Pourhabibi, K.-L. Ong, B.H. Kam, Y.L. Boo, Fraud detection: A systematic literature review of graph-based anomaly detection approaches, Decision Support Systems 133 (2020), https://doi.org/10.1016/j.dss.2020.113303 113303.
[4] D. Chen, P. Wang, L. Yue, Y. Zhang, T. Jia, Anomaly detection in surveillance video based on bidirectional prediction, Image and Vision Computing 98 (2020), https://doi.org/10.1016/j.imavis.2020.103915 103915.
[5] G. Pang, C. Yan, C. Shen, A. van den Hengel, X. Bai, Self-trained deep ordinal regression for end-to-end video anomaly detection, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vol. 1, IEEE Computer Society, Los Alamitos, CA, USA, 2020, pp. 12170–12179. doi:10.1109/CVPR42600.2020.01219..
[6] T. Nakao, S. Hanaoka, Y. Nomura, M. Murata, T. Takenaga, S. Miki, T. Watadani, T. Yoshikawa, N. Hayashi, O. Abe, Unsupervised deep anomaly detection in chest radiographs, Journal of Digital Imaging (2021) 1–10. doi:10.1007/s10278-020-00413-2.
[7] J. Jang, H.H. Lee, J.-A. Park, H. Kim, Unsupervised anomaly detection using generative adversarial networks in 1h-mrs of the brain, Journal of Magnetic Resonance 325 (2021), https://doi.org/10.1016/j.jmr.2021.106936 106936.
[8] Y. Tian, G. Maicas, L.Z.C.T. Pu, R. Singh, J.W. Verjans, G. Carneiro, Few-shot anomaly detection for polyp frames from colonoscopy, in: Medical Image Computing and Computer Assisted Intervention – MICCAI 2020, Springer International Publishing, Cham, 2020, pp. 274–284.
[9] B. Schölkopf, J.C. Platt, J. Shawe-Taylor, A.J. Smola, R.C. Williamson, Estimating the support of a high-dimensional distribution, Neural Computation 13 (7) (2001) 1443–1471, https://doi.org/10.1162/089976601750264965.
[10] E. Parzen, On estimation of a probability density function and mode, The Annals of Mathematical Statistics 33 (3) (1962) 1065–1076.
[11] D.M. Tax, R.P. Duin, Support vector data description, Machine Learning 54 (1) (2004) 45–66, https://doi.org/10.1023/B:MACH.0000008084.60811.49.
[12] G.E. Hinton, R.R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science 313 (5786) (2006) 504–507. doi:10.1126/science.1127647..
[13] M. Nicolau, J. McDermott, et al., Learning neural representations for network anomaly detection, IEEE Transactions on Cybernetics 49 (8) (2018) 3074–3087, https://doi.org/10.1109/TCYB.2018.2838668.
[14] A. Borghesi, A. Bartolini, M. Lombardi, M. Milano, L. Benini, Anomaly detection using autoencoders in high performance computing systems, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, 2019, pp. 9428–9433. doi:10.1609/aaai.v33i01.33019428..
[15] J. Chow, Z. Su, J. Wu, P. Tan, X. Mao, Y. Wang, Anomaly detection of defects on concrete structures with the convolutional autoencoder, Advanced Engineering Informatics 45 (2020), https://doi.org/10.1016/j.aei.2020.101105 101105.
[16] J. Kolberg, M. Grimmer, M. Gomez-Barrero, C. Busch, Anomaly detection with convolutional autoencoders for fingerprint presentation attack detection, IEEE Transactions on Biometrics, Behavior, and Identity Science 3 (2) (2021) 190–202, https://doi.org/10.1109/TBIOM.2021.3050036.
[17] D.P. Kingma, M. Welling, Auto-encoding variational bayes, in: 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14–16, 2014, Conference Track Proceedings, 2014..
[18] J. An, S. Cho, Variational autoencoder based anomaly detection using reconstruction probability, Special Lecture on IE 2 (1) (2015) 1–18.
[19] H. Khalid, S.S. Woo, Oc-fakedect: Classifying deepfakes using one-class variational autoencoder, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2020..
[20] B. Zong, Q. Song, M.R. Min, W. Cheng, C. Lumezanu, D. Cho, H. Chen, Deep autoencoding gaussian mixture model for unsupervised anomaly detection, in: International Conference on Learning Representations, 2018..
[21] X. Wang, Y. Du, S. Lin, P. Cui, Y. Shen, Y. Yang, advae: A self-adversarial variational autoencoder with gaussian anomaly prior knowledge for anomaly detection, Knowledge-Based Systems 190 (2020), https://doi.org/10.1016/j.knosys.2019.105187 105187.
[22] A.L. Alfeo, M.G. Cimino, G. Manco, E. Ritacco, G. Vaglini, Using an autoencoder in the design of an anomaly detector for smart manufacturing, Pattern Recognition Letters 136 (2020) 272–278, https://doi.org/10.1016/j.patrec.2020.06.008.
[23] Z. Wang, Y.-J. Cha, Unsupervised deep learning approach using a deep auto-encoder with an one-class support vector machine to detect structural damage, Structural Health Monitoring (2020), https://doi.org/10.1177/1475921720934051, 1475921720934051.
[24] C. Ieracitano, A. Adeel, F.C. Morabito, A. Hussain, A novel statistical analysis and autoencoder driven intelligent intrusion detection approach, Neurocomputing 387 (2020) 51–62, https://doi.org/10.1016/j.neucom.2019.11.016.
[25] K.R.P.M. Dutta V., Choras M., Hybrid model for improving the classification effectiveness of network intrusion detection., in: 13th International Conference on Computational Intelligence in Security for Information Systems (CISIS 2020), 2019..
[26] R. Yao, C. Liu, L. Zhang, P. Peng, Unsupervised anomaly detection using variational auto-encoder based feature extraction, in: 2019 IEEE International Conference on Prognostics and Health Management (ICPHM), IEEE, 2019, pp. 1–7, https://doi.org/10.1109/ICPHM.2019.8819434.
[27] Z. Ghrib, R. Jaziri, R. Romdhane, Hybrid approach for anomaly detection in time series data, in: 2020 International Joint Conference on Neural Networks (IJCNN), 2020, pp. 1–7, https://doi.org/10.1109/IJCNN48605.2020.9207013.
[28] R. Chalapathy, A.K. Menon, S. Chawla, Anomaly detection using one-class neural networks, arXiv preprint arXiv:1802.06360..
[29] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S.A. Siddiqui, A. Binder, E. Müller, M. Kloft, Deep one-class classification, in: International Conference on Machine Learning, 2018, pp. 4393–4402.
[30] Y. LeCun, C. Cortes, C. Burges, Mnist handwritten digit database..
[31] A. Krizhevsky, G. Hinton, et al., Learning multiple layers of features from tiny images..
[32] J. Stallkamp, M. Schlipsing, J. Salmen, C. Igel, The german traffic sign recognition benchmark: a multi-class classification competition, in: The 2011 International Joint Conference on Neural Networks, IEEE, 2011, pp. 1453–1460, https://doi.org/10.1109/IJCNN.2011.6033395.
[33] F.T. Liu, K.M. Ting, Z.-H. Zhou, Isolation forest, in: 2008 Eighth IEEE International Conference on Data Mining, IEEE, 2008, pp. 413–422, https://doi.org/10.1109/ICDM.2008.17.
[34] J. Masci, U. Meier, D. Cireşan, J. Schmidhuber, Stacked convolutional auto-encoders for hierarchical feature extraction, in: International Conference on Artificial Neural Networks, Springer, 2011, pp. 52–59, https://doi.org/10.1007/978-3-642-21735-7_7.
[35] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, A. Madry, Adversarial examples are not bugs, they are features, in: Advances in Neural Information Processing Systems, vol. 32, Curran Associates Inc, 2019..



Yu Zhou received the B.S., M.S. and Ph.D. degrees in electrical engineering from Xidian University, Xi'an, China, in 2001, 2004 and 2011, respectively. He is currently an Associate Professor with the National Laboratory of Radar Signal Processing, Xidian University. His research interests include ground moving target indication, spectrum sharing and radar automatic target recognition.

Xiaomin Liang received the B.S. degree in electronic engineering from Xidian University, Xi'an, China, in 2018. She is currently pursuing the M.S. degree in signal and information processing with Xidian University, Xi'an, China.

Wei Zhang received B.S. and Ph.D. degrees in electrical and information engineering from Xidian University, Xi'an China, in Jul. 2014 and Dec. 2020 respectively. His research interests include radar signal processing, radar automatic target recognition, and pattern recognition.

Linrang Zhang received the M.S and Ph.D. degrees in electrical engineering from Xidian University, Xi'an, China, in 1991 and 1999, respectively. He is currently a Full Professor with the National Laboratory of Radar Signal Processing, Xidian University. He has authored or coauthored three books and published over 100 papers. His research interests include radar system analysis and simulation, radar signal processing, and jamming suppression.

Xing Song received the B.S. degree from Northeast Electric Power University, Jilin, China, in 2019. He is currently pursuing the M.S. degree in signal and information processing with Xidian University, Xi'an, China.