GANAD:A GAN-based method for network anomaly detection

Jie Fu
Wuhan University
Lina Wang (✉ lnwang@whu.edu.cn)
Wuhan University
Jianpeng Ke
Wuhan University
Kang Yang
Wuhan University
Rongwei Yu
Wuhan University

Research Article

Keywords: Network anomaly detection, WGAN, Gradient penalty, Spectral normalization

Posted Date: September 22nd, 2022

DOI: https://doi.org/10.21203/rs.3.rs-2081269/v1

GANAD:A GAN-based method for network anomaly detection

Jie Fu[1], Lina Wang[1*], Jianpeng Ke[1], Kang Yang[1] and Rongwei Yu[1]

[1]Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Cyber Science and Engineering, Wuhan University, Wuhan, 430072, China.

*Corresponding author(s). E-mail(s): lnwang@whu.edu.cn; Contributing authors: whuerfu@whu.edu.cn; kejianpeng@whu.edu.cn; kang_yang@whu.edu.cn; roewe.yu@whu.edu.cn;

Abstract

Cyber-intrusion can lead to severe threats to network, i,e., system paralysis, information leaky, and economic losses. To protect network security, anomaly detection methods based on generative adversarial networks (GAN) for hindering cyber-intrusion have been proposed. However, existing GAN-based anomaly score methods built upon the generator network are designed for data synthesis, which would get unappealing performance on the anomaly detection task. Therefore, their low-efficient and unstable performance make detection task still quite challenging. To cope with these issues, we propose a novel GAN-based approach **GANAD** to address the above problems which is specifically designed for anomaly identification rather than data synthesis. Specifically, it first proposed for a similar auto-encoder architecture, which makes up for the time-consuming problem of the traditional generator loss computation. In order to stabilize the training, the proposed discriminator training replace JS divergence with Wasserstein distance adding gradient penalty. Then, it utilizes a new training strategy to better learn minority abnormal distribution from normal data, which contributes to the detection precision. Therefore, our approach can ensure the detection performance, and overcome the problem of unstable in the process of GAN training. Experimental results demonstrate that our approach achieves superior performance than state-of-the-art methods and reduces time consumption at the same time.

Keywords: Network anomaly detection, WGAN, Gradient penalty, Spectral normalization

1 Introduction

At present, the Internet and computer networks has realized the interconnection of information, accelerated the speed of information transmission, and changed the way of data transmission. However, the widespread adoption of computer networks introduces several security threats that may cause misbehavior and severe damage. Those threats often keep changing and will evolve to new unknown variants [1]. To prevent these threats, anomaly detection technologies are employed by a classification engine that can determine the safety of the network. Nowadays, an excellent anomaly detection system is required to discover various anomalies with new network attacks emerging efficiently. Recently there are plenty of data-driven network intrusion detection, which has a tendency towards minority attack classes compared to normal traffic [2]. Firstly, many supervised methods works have been proposed successively which classify behaviors that do not match the normal behavior as attacks. Representative supervised methods such as decision tree (DT) [3], support vector machine (SVM) [4] could analyze and identify these behaviors successfully. However, they were also shown to not scale to the large real-world network data sets which the amount of attack traffic in the network is limited. Therefore, unsupervised and weakly-supervised methods like REPEN [5], PRO [6], and semi-supervised methods like DeepSAD [7],capable of classifying anomalies without labeled data, were deemed for defending anomaly threats. However, they failed to detect all abnormal behaviors efficiently because of unknown anomalies or data contamination etc. Therefore, these methods can not be capable of handing the current's cyber anomaly threats, level of sophistication, and flexible.

Moreover, the lack of prior knowledge, i.e, the attack Categories (Zero-day attack) is a important challenge in the detection task, as they need to be detected quickly to be avoided great damage. On the one hand, many network infrastructures and individual devices within CPSs or IoT have unknown vulnerabilities, which complicate the security solutions. On the other hand, with the 5G network and Cloud Services fast development, the transmission and bandwidth of cyber-attack traffic will significantly increase, and thus these intrusions may be difficult to be detected in real time and stably. However, GAN [8] has been proposed in this field and achieved excellent performance on complex network traffic data sets. It is well known that unknown network intrusions will also behave a pattern more similar to a known anomaly pattern rather than the normal data [9]. Since GAN is able to learn implicit probability distribution, the discriminator can find the generated or fake samples. Recent work like AnoGAN [10] is the first GAN-based method, which extracts normal samples features to discriminate anomalies. GANomaly [11] further improves the generator over the previous work by utilizing an encoder-decoder-encoder network to change the generator network.

High efficiency and continuous stabilization of detection can be challenging for current methods. Existing GAN-based anomaly detection methods can not satisfy the low-latency and stable detection requirement of IDS. In addition, these work above are mainly focused on data synthesis of generator [12] and have obtained suboptimal anomalous scores for intrusion detection. Thus, two main challenges are yet to be addressed. On the one hand, the previous GAN-based methods get poor performance due to solely relying on the generator. On the other hand, because of optimization problem which find a latent variable, extent GAN-based methods [10], [13] cannot well solve the problems that both satisfy stable training and efficiency, hence they may fail to identify the anomalies in real-time.

To overcome the above hurdles, we design a WGAN-based model called **GANAD** that improves its performance by using an improved GAN network structure and is applied flexibly. Our architecture models input data as network heterogeneous node and the GAN as similar auto-encoder networks to handle character features of data. It introduces a new designed encoder with layers using spectral normalization that can facilitate generator to generate samples with more wider variety. Besides, This design can accelerate the residual loss computation by avoiding the use of typical GAN structure [13] which needs find corresponding latent space iteratively, and increase the efficiency of detection by eliminating the need for optimization problem of generator. And the proposed network allows discriminator and generator training with spectral normalization, which can capture important hidden information in the sample distribution even with little overlap between samples. Moreover, we improve the architecture [14] by replacing GAN discriminator with no restrain, required for computing discrimination loss, for a discriminator trained with gradient penalty, which stabilize the adversarial training. Therefore, it not only can simulate more accurate data distribution to get superior performance but also reduce computational cost. Furthermore, our proposed approach utilizes residual loss and discrimination loss to construct a training strategy for modeling weak abnormal supervisory signal. At last, our work achieves a equilibrium between optimum detection accuracy and efficient performance. Three different data sets are conducted to verify the effectiveness and generalization of our approach, and experimental results demonstrate that our approach outperforms the state-of-the-art methods.

The main contributions of this paper are summarized as follows:

- We propose an anomaly based IDS for network anomaly detection using improved GAN, called GANAD, which can achieve efficient intrusion detection. Experiment shows that the novel network architecture can make our approach get optimal performance and overcome the challenge of lack of prior knowledge.

- Proposal of a novel and faster method for computing the discrimination and reconstruction loss can improve the detection performance, which can meet trade-off of the high efficiency and stable requirement.
- To get better evaluation scores, we propose a novel training strategy to model abnormal weakly labeled data space between majority normal and minority abnormal samples, and also between the real samples and generated samples.
- The proposed GANAD is validated on three real-world network datasets for binary classification and mutil classification tasks. Experimental results demonstrate that the proposed approach is superior to the state-of-the-art network anomaly detection approaches, achieving both stabilization and efficiency.

2 Related work

We have surveyed many research efforts and encouraged progress on network anomaly detection. In this section, we briefly introduce existing works in this field according to traditional work, DNN-based work, and GAN-based work.

2.1 Traditional work

In the initial stage, there were traditional methods, and most of them utilized the supervised machine learning algorithm. For example, the distance-based method [15] was applied to evaluate whether the data is anomalous by using the distances of nearest neighbors or clusters in the data. Clustering-based approaches have also been proposed. Blowers et al. [16] proposed a method called DBSCAN to identify anomalies in the network. Next, Khan et al. [17] proposed a method which use genetic algorithm to detect anomalies. Shone et al. [18] discriminated anomalies using random forest as a classifier. To multi classify various anomalies, Snehal et al. [19] proposed that combining SVM and decision tree is to build a multi classification anomaly detection system which construct multi classification SVM by using binary classification tree. Selvakumar et al. [20] proposed a fuzzy and rough set based nearest neighborhood algorithm (FRNN) to classify network trace dataset. Representative marching learning methods based on density evaluation like Local Outlier Factor (LOF) [21], Robust Covariance [22] and Isolated Forests (IF) [23], which can solve the problem of too little labeled data to some extent. Due to the rapid development of new network, network traffics having ultra-high dimensions are ubiquitous which make these methods ineffective.

2.2 DNN-based work

More recent works were based on deep neural networks (DNN), DNN-based algorithms have also been widely used in network anomaly detection. In the initial stage, Ingre et al. Recurrent neural network (RNN) was used by Torres et al. [24] to capture the temporal features of network data. Deng et al. [25] combined a structure learning approach with graph neural networks, additionally using attention weights to provide explainability for the detected anomalies. Kwon et al. [26] established three different Convolution Neural Network (CNN) architectures based on structural scalability to improve network anomaly detection performance. In another study on the same task, Zhao et al. [27] suggested that a network intrusion detection framework use DBN and probabilistic neural network. This method demonstrated that the effect of combination was better than that of the non-optimized DBN. In the next stage, Pang et al. [28] adopted a reinforcement learning method called DPLAN that optimizes learning of marked abnormal data and unmarked abnormal data to identify unknown anomalies. Wang et al. [29] proposed an unsupervised representation learning method called RDP that learns data distance in a random project space by training a neural network with random mapping. Pang et al. [30] proposed a method called DevNet that realize abnormal score learning by using neural deviation learning, and optimized the representation of abnormal score by integrating neural network, Gaussian priori and Z-Score-based deviation loss function. Autoencoder (AE) [31], variational autoencoder (VAE) [32] and deep auto-encoding Gaussian mixture model (DAGMM) [33] have been successively used for the purpose of abnormal data detection. But these methods model the data distribution and derive anomaly scoring criteria based on Gaussian mixture. In a follow-up study, Zhai et al. [34] proposed an energy-based model DSEBM, using the accumulated energy between the class denoising autoencoder layers to obtain the anomaly score. Lately, Mirsky et al. [35] proposed a method called Kitsune, a plug and play network intrusion detection system (NIDS) which can learn to detect attacks on the local network, without supervision.

2.3 GAN-based work

Finally, Generative Adversarial Networks (GAN) were applied to network anomaly detection. The common practice of them was to determine whether the test sample is in an abnormal state by measuring the discreteness between the test sample distribution and the learning distribution. AnoGAN [10] generated real space samples from the latent space, then defined abnormal score based on the discrepancy between the generated samples obtained by latent space and the test samples. In addition, this method optimized the update of the generated network iteratively via the back propagation algorithm. This iteration optimization process is calculating complexity and time-consuming which is not applicable to real-time network anomaly detection. Instead of utilizing a typical GAN, Efficient GAN-Based Anomaly Detection (EGBAD) [36] first brings the BiGAN architecture to the anomaly detection domain. Lately, ALAD [37] adopted bi-directional GANs that simultaneously learn an encoder network during training. However, this design avoids the computational expensive in inference procedure, its discriminator training is still time-consuming at test time. Because successes of GAN in generating realistic complex datasets, MAD-GAN [13] used the Long-Short-Term-Memory Recurrent Neural Networks (LSTM-RNN) as the base models in the GAN framework to capture the temporal correlation of time series distributions. Mohammadi [38] proposed an end-to-end deep architecture for IDS using Generative Adversarial Networks (GANs) for training deep models in semi-unsupervised setting. f-AnoGAN [39] improved the computational efficiency by adding an encoder before the generator to map from data to the latent space. In addition, this method proposed three different architectures for mapping samples to the latent space. However, the above work lacked an evaluation on the time cost of these architectures. Recently, generative adversarial networks (GANs) as a promising unsupervised approach to detect cyber-attacks, FID-GAN [40] was a unsupervised intrusion detection system (IDS) for cyber–physical systems which was proposed for a fog architecture achieving higher detection rates. The work IGAN [41] tackled the class imbalance problem by generating new representative instances for minority classes with an imbalanced data filter and convolutional layers to the typical GAN. ACGAN [42] proposed an auxiliary classifier generative adversarial network to generate synthesized samples to augment the ID datasets.

3 Proposed method

3.1 GAN with MLP

GANs as a powerful modeling frameworks, it is suitable to deal with high-dimensional data like traffic samples. Designed by game theory, GANs consist of two adversary networks: a generator G and a corresponding discriminator D. The generator network plays a role in producing synthetic data samples which are similar to real sample patterns from a random latent space. In addition, the discriminator network plays a role in distinguishing generated or real samples. Following a typical GAN framework, the synthetic samples generated by generator as the inputs are passed to the discriminator, which will try to find the generated (i.e. "fake") data samples from the actual (i.e. "real") normal training data samples. The two models of GANs are trained together in a zero-sum adversarial minimax game, in which the generator tries to maximize the probability of producing outputs recognized as real, while the discriminator tries to minimize the same probability. Therefore, they can be regarded as two agents playing a minimax game with value function $V(G, D)$ as follows:

$$\begin{aligned}\min_{G}\max_{D} V(D,G) =& \mathcal{E}_{x\sim p_{\text{data}}\,(X)}\left[\log D(x)\right] \\ &+ \mathcal{E}_{z\sim p_{z}(Z)}\left[\log(1-D(G(z)))\right]\end{aligned} \tag{1}$$

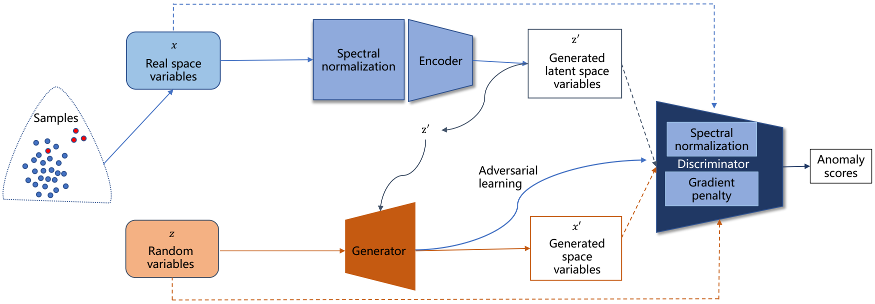


Fig. 1 Proposed GANAD: GAN-based anomaly detection

Because of the heterogeneous network, the real-world network traffic data is diversify and complexity. In order to handle these high-dimensional and diversify data, the discriminator and generator are constructed as MLP networks. We asume that network traffic data samples are not independent of each other and there is a unseen relationship among them. Thus, the each layers of this network will plays a important role in capturing the non-liner and combination features of network data. In our framework, a single MLP network as one small part of GAN is to obtain the correlations characters among the data, which can be prepared for the detection task.

3.2 Network Architecture and Encoder

Referring to the recently developed GAN network architecture, especially BiGAN proposed by Donahue [14] has one more encoder to map the real samples to the latent space state. Hence there is no need to find the latent state again corresponding to the samples in the test process. This design saves time by avoiding the use of back-propagation algorithms. Inspired by the computational efficiency of BiGAN, we build a GAN framework that map the input data samples to the latent space through the encoder network during training. Our model improves its latent representation ability of data and testing efficiency by adding spectral normalization into the encoder network. The overall architecture of our model framework is shown in Figure 1. Where x represents real space variables, z is the random variables sampled from a latent distribution. z' is the generated latent space variable obtained by the encoder, x' is the new space variable generated by the generator. It consists of three main parts, the encoder, generator, and discriminator. Firstly, the real data samples are preprocessed to obtain x. Then x and z are input to the encoder and generator networks respectively to obtain a accurate latent distribution of z' as well as a reconstructed generated distribution of x' .

The discriminator D use Xavier initializer to initialize the weights matrix, and is trained with the constrain gradient penalty to stabilize the discrimination training. Moreover, it is trained with Adam optimizer to minimize the earth-mover distance between its predictions and real labels. Its loss is presented as Eqs.(2):

$$L_D = \sum_{i=1}^{n} \left[E\left[D(G(z_i)) \right] - E\left[D(x_i) \right] + \theta E\left[(\|\nabla_{\hat{x_\mathrm{i}}} D(\hat{x_i})\|_2 - 1)^2 \right] \right] \qquad (2)$$

where n is the number of samples, x_i $\forall i \in \{1, \ldots, n\}$ are one of training data samples, which should be distinguished as real and recognized as normal samples by our network. And z_i are the latent space samples, they should be distinguished as fake and detected as anomalies by discriminator network. In addition, we define θ is the penalty coefficient, which is used to enforce the unit gradient norm constraint. $\hat{x_i}$ is random samples sampled uniformly along straight lines between pairs of points sampled from the data distribution and generator distribution. The weights of generator G are also initialized with Xavier initializer, it is trained with Adam optimizer to minimize the Wasserstein-1. And its objective is to fool the discriminator into wrong decision recognizing the generated samples as real. Its loss value function is given by:

$$L_G = \sum_{i=1}^{n} \left[E\left[D(G(z_i)) \right] \right] \qquad (3)$$

In standard GAN architecture, the discriminator D is always used to discriminate real and generated samples. However, discriminator playing fundamental role of performance not well, [10] indicates that generator reconstructed sample can be used to localize the anomalous distribution in classification tasks. Therefore, our method will adopt a new strategy to detect minority abnormal samples with a new designed GAN by computing an abnormal score through the convex combination of reconstruction loss and discriminator loss. The reconstruction loss measures the dissimilarity between the evaluated real sample and the generated sample in the input domain space, while the discriminator loss takes into account the discriminator network output. In the adversarial training phase, generator learned an implicit representation of evaluated sample always affects the discriminator decision. Thus, the reconstruction loss is very important since it can be used to measure the probability of an evaluated sample being an anomaly sample.

As we all known that it is first necessary to find corresponding sample representation being evaluated in the latent space for computing reconstruction loss L_R. The literature [13] has shown that computing L_R is time-consuming through the inversion of the generator. In order to compute L_R more fast, [40] proposed a encoder mapping from the data pattern space to the latent space directly, which use the auto-encoder to train the proposed encoder [43]. For this purpose, our architecture builds a new designed encoder that maps random data patterns to the latent space. In contrast to [40] that train encoder

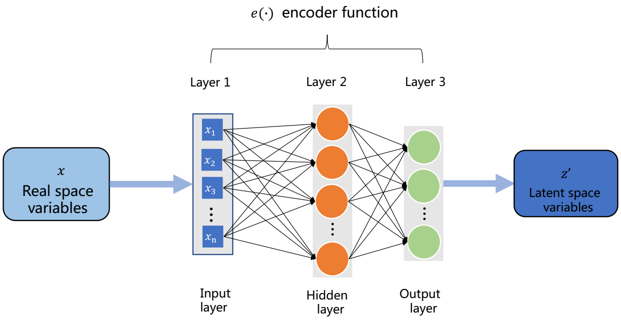


Fig. 2 The architecture of Encoder

through auto-encoder, our encoder trained with a simple MLP network which results in good performance is more suitable for detecting emergency intrusion.

In order to compute L_R efficiently, we utilize the simple MLP network rather than auto-encoder to train **encoder** E. We train an encoder that obtains the latent space representations of real data patterns by mapping data patterns to the latent space. The proposed encoder E is introduced by Figure 2. In addition, we train the generator as the decoder part of autoencoder, which is to ensure that x and corresponding $G(E(z))$ are as similar as possible. Figure 3 shows the relationship between the encoder and generator space mappings. To stabilize the training of the network, spectral normalization (SN) [44] is applied to normalize the weight matrix of the full connection layer of the encoder. Compared with the encoder of [40], our encoder with spectral normalization is able to learn the latent representation of the optimal data distribution. Moreover, the encoder is trained by measuring the euclidean distance between the he input data x and reconstructed data $G(E(z))$ as follow function:

$$L_R = \sqrt{\sum_{i=1}^{n} \left[x_i - G\left(E\left(x_i\right)\right)\right]^2} \tag{4}$$

where n is the data dimension.

3.3 System Model

The architecture of our proposed system model showed in figure 4 is based on the MLP framework and deployed in three parts: 1) input part; 2) GAN part; 3) anomaly score part. All of them are full connected layers. The input include training data, testing data and random latent space. The Training data samples are the normal data patterns used to train the GAN and the encoder. The testing data patterns that are evaluated by our system model. The GAN part is equipped with the discriminator, generator and encoder. On the left

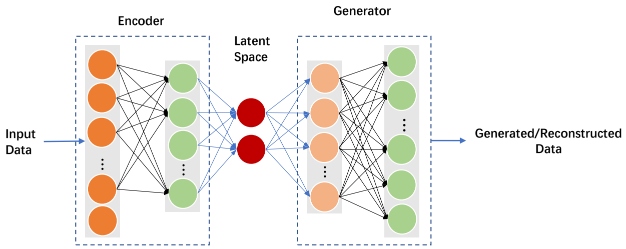


Fig. 3 The encoder and generator

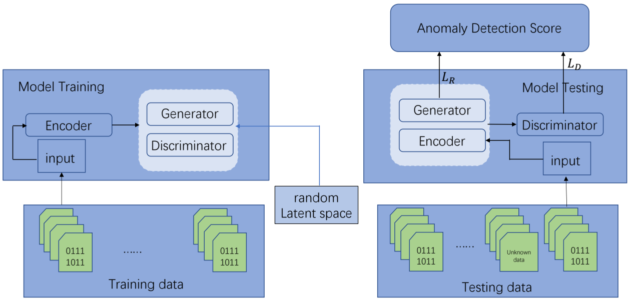


Fig. 4 The complete framework of our system model

is a GAN framework in which the generator and discriminator are obtained with iterative adversarial training. Then the encoder is trained within the MLP architecture while using the trained generator as the decoder. On the right, the input part send unknown data patterns and real data patterns to be evaluated by the system. On the top, according to the anomaly score computed discrimination and reconstruction losses by anomaly detection system, we can decide whether the evaluated pattern is an anomaly or not.

The discriminator network D is the other part of the whole architecture, and it is, with the generator part and encoder part, used to build the our GAN architecture. However, even many modified loss functions proposed can misbehave in the presence of a good discriminator [45]. Specially, real-word network traffic sample quality is always not well, since the abnormal data samples is minority and unusual. Thus, WGAN value function appearing to correlate with sample quality isn't making optimization of the generator easier, which results in the undesired detection performance. Unlike other approaches that directly minimizes the training loss value function, adding gradient penalty constrain to our discriminator is a better solution to optimize the adversarial training. Gradient penalty term is a model-level constraint that does not affect the ability of the neural network learning. Hence it allows the discriminator to approximate the Wasserstein metric more accurately. The input layer is where the data pairs $(x, E(x))$ and $(z, G(z))$ are input. The pairs of data patterns as input which can contain more hidden information promote the discriminator detection performance. In addition, spectral normalization is applied on

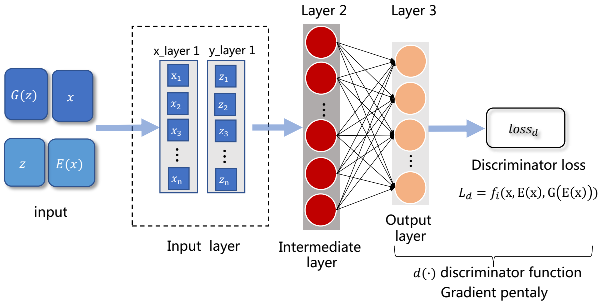


Fig. 5 The architecture of discriminator

the input layers which not only stabilizes the training of the discriminator but also improves the performance of the discriminator by re-parameterization. Moreover, it can also update the weight of the hidden layer, which be used to compensate for the lack of gradient penalty using in a multi-category real sample scenario applications [46].

Figure 5 describes the architecture of discriminator, Where x, $G(z)$, $E(x)$ and z are input variables, x is the data samples at a real distribution. z represents the latent space, $G(z)$ is the generated data samples obtained by generator, $E(x)$ can generate new latent distribution. Firstly we input z and $E(x)$ into the x_layer while input x and generated latent space $G(z)$ into y_layer to get the vectors which is input into the intermediate layer. Spectral normalization is adding to each of the two input layers. The intermediate layer is embedded in discriminator, which help to better evaluate the difference between the pairs of discriminator's input. This layer is used to soften the decision of discriminator to obtain a more moderate result, and also used to be computing loss value as feature matching. Finally, the output layer is trained by the discriminator loss function $d(\cdot)$ to obtain the final loss value. Then the output is obtained through the intermediate layer.

3.4 GANAD Training

3.4.1 training strategy

In this paper, GANAD is an anomaly detection approach based on the discriminator that evaluates how different a sample distribution is from other data. We introduce a WGAN-based GAN to simulate data distribution precisely, then define anomaly scores of all data samples by quantifying differences of distributions between real samples and generated samples. Finally, we discriminate the anomalies from normal samples through a testing criterion.

To this end, we first need to model the data distribution accurately: This means that the generator is used to learn the normal data distribution until the generated data distribution approximates the normal data distribution: $p_G(x) \approx p_X(x)$. Our model can obtain better latent space representation by the encoder, thus it is able to restructure real distribution precisely. In addition, generator can better simulate the true distribution by the adversary learning.

In this context, we should redefine an novel anomaly score that measures the attributes of a data sample. Generally, GAN learns the latent feature space of the data by generator network, and determines whether it is abnormal or not by calculating the normal probability obtained from the test samples. So, the residual form between real example and generated example is generally defined as the anomaly score. Unlike the above define score, there are reconstruction loss and discriminator loss at the last update iteration of the mapping procedure to the latent space respectively in the testing stage, defined by us as scores which constitute the anomaly score. One of them is reconstruction score $\mathcal{R}_{rec}$: We adopt the generator to measure the dissimilarity between the generated samples and the real samples in the real space. The other one is discriminator score $\mathcal{D}_d$: We determines the dissimilarity between the generated samples and the real samples during adversarial training. Inspired by [10], we use the convex combination of reconstruction error and discriminator error to judge whether the sample is abnormal or not. Therefore, the abnormal score in this paper is designed as shown in the Eqs.(5):

$$\text{Score} = \alpha \mathcal{R}_{rec} + (1 - \alpha)\mathcal{D}_d \tag{5}$$

Where α is a constant that varies between 0 and 1, the reconstruction score $\mathcal{R}_{rec}$ and the discriminator score $\mathcal{D}_d$ are defined by the reconstruction loss $\mathcal{L}_R$ and the discriminator loss $\mathcal{L}_D$ respectively.

$$\mathcal{L}_R = |x - G(E(x))| \tag{6}$$

Here, we define $\mathcal{L}_R$ as the reconstruction loss function as shown in Eqs.(6), a cost function based on the feature space specifically is used to measure the variability between the test and generated samples. $G(E(x))$ denotes samples reconstructed from the latent space corresponding to x. The encoder and generator collaborate with each other to reconstruct the input. Then the input data is passed through the encoder and generator to get the output. There is a reconstruction error between the input and output. $\mathcal{L}_D$ represents the discriminator loss function, we have two expressions for it. As shown in Eqs.(7), the first is that we use the cross-entropy loss function δ to represent the difference between source representation of real samples x and latent representation of samples $E(x)$. Next, Eqs.(8) shows that we use the feature matching loss to define our $\mathcal{L}_D$. This evaluates if the reconstructed data has similar features in the discriminator as the true sample.

$$\mathcal{L}_{D1} = \delta(x, E(x)) \tag{7}$$

$$\mathcal{L}_{D2} = f_i(x, E(x), G(E(x))) \tag{8}$$

In the discriminator loss function, f_i is the intermediate layer of the discriminator network. $f_i(\cdot)$ is the output of this intermediate layer. Specifically, $f_i(\cdot)$ plays a important role in training procedure which maps the space of data to the feature space. Generally, when the dataset size is relatively large, multiple middle layers will help to better evaluate the difference between a pair of discriminator inputs by features. In our network, we only use one layer. With the addition of an intermediate layer where we apply L1 regularization to auxiliary main function, this enables us to capture the rich feature information of the sample.

For binary classification task, we only need to distinguish whether the sample is an anomaly or normal and define the reconstruction error value of the sample as a score. As shown in Eqs.(9), we propose a cost function $\mathcal{L}_r$ to identify reconstructed sample $E(x)$ from x. Generally, the cross-entropy loss is adopted to train generator as classifier. Then we obtain the residual value between them by this loss. In order to simulate weak anomaly supervised signal over data distribution, we utilize the following objective function Eqs.(10) to enable the generator to generate data samples that match the statistics of real data. Generally, the standard cross-entropy loss function is used to enable the discriminator to correctly distinguish the real samples from the generated samples. However, for multi classification task, feature matching loss function is good at improving the performance of GAN training in our work than other loss functions.

$$\mathcal{L}_r = \mathcal{L}_{D1} \tag{9}$$

$$\mathcal{L}_{fm} = \mathcal{L}_{D2} \tag{10}$$

Our goal is to obtain precise anomalous data distribution that is used to identify various anomalies. To achieve this, we use the following objective function Eqs.(11) to train the discriminator as a classifier. When it is for binary classification task, $\lambda = 1$, otherwise, $\lambda = 0$.

$$\mathcal{L}_D = \lambda\mathcal{L}_r + (1-\lambda)\mathcal{L}_{fm} \tag{11}$$

3.4.2 Model training

To make our model training more stable, we make a series of improvements that utilize a new network structure and training strategy. Inspired by SN, we apply SN in the discriminator network to constrain the Lipschitz limitation to reach the saddle point of the discriminator-based loss function. In addition, discriminator trained with SN allows the parameter matrix to use as many features as possible for discrimination work while satisfying local 1-Lipschitz constraint. Unlike Wasserstein distance-based GAN (WGAN) [47] which directly adopts weight-clipping to deal with 1-Lipschitz weight constraint, we adopt gradient penalty [45] as the gradient regulization method to solve gradient explosion or disappearance. Here the saddle point problem $\min_{G,E} \max_D V(D,E,G)$ includes the gradient regularization $V_{gr}(D)$ on the discriminator, and spectral normalization $V_{sn}(D,E)$ on the discriminator and encoder. Eqs.(12) defined below solves our saddle-point problem.

$$V(D,E,G) = V_w(D,E,G) + V_{gr}(D) + V_{sn}(D,E) \tag{12}$$

We deal with the objective function by fining-tune the model training as show in Eqs.(13)and Eqs.(14):

$$\min_{G,E} \max_{D} V(D,E,G) \tag{13}$$

$$\begin{aligned} V(D,E,G) = \; & \mathbb{E}_{x \sim pX} \left[\mathbb{E}_{z \sim pE(z|x)} D(x,z)_{w-gp} \right] \\ & + \mathbb{E}_{z \sim pZ} \left[\mathbb{E}_{x \sim pG(x|z)} \left[1 - \| D(x,z) \|_w \right] \right] \end{aligned} \tag{14}$$

Where D, E, and G respectively represents the discriminator, encoder, and generator, pX represents the distribution of data samples, and pZ is the distribution over the latent space. $pE(z \mid x)$ and $pG(x \mid z)$ are the joint data distribution learned by encoder and generator respectively. $w - gp$ represents Wasserstein distance with gradient penalty term, which constrains the hyper-parameters to satisfy the 1-lipschitz continuity. w is Wasserstein distance. In our model, we apply $w - gp$ to train the discriminator D while w is used to train the encoder and generator. Therefore, the improved discriminator loss function solves two problems existing in WGAN by setting an additional gradient penalty mechanism, which are the concentration of parameters and gradient disappearance or explosion caused by gradient clipping. E used on the encoder is a non-linear parametric function in the same way as G, and it can be trained using Wasserstein distance.

4 Experiments

In this section, we conduct experiments on three real-world datasets to validate the effectiveness of the proposed approach. It includes four parts, datasets processing, simulation experiments, comparison algorithms, detection performance, and ablation studies.

4.1 Datasets processing

To evaluate the performance of our model, we run experiments including binary classification, multi classification on KDDCUP'99 (10 percent) [48], NSL-KDD [48] and UNSW_NB15 [49] benchmark datasets. KDDCUP'99 (10 percent) is a dataset widely used for the testing of network anomaly detector, which is

Algorithm 1 GAN-based adversarial learning network anomaly detection

```
input: $x$, real space variables; $z$, latent space variables; $E$, encoder function; $G$, generator function; $f$, the feature layer of $D$.
Output: $S(x)$, the anomaly score about each sample.
Input $x$
while number of training iterations do
    for generation-steps do
        $z' \leftarrow E(x)$, Encoder samples
        $x' \leftarrow G(z)$, Reconstruct samples
        $\bar{W}_{sn}(W_E) := W/\partial(W)$, updating weights of encoder
        $\bar{W}_{sn}(W_D) := W/\partial(W)$, updating weights of discriminator
        when training reach stable
    end for
    for detection-steps do
        Procedure inference
        $\mathcal{L}_R \Longleftarrow |x - G(z')|$
        $\mathcal{L}_D \Longleftarrow f_x(x, z)$
        if binary classify then
            $\mathcal{L}_D = \mathcal{L}_r = \delta(z', G(z'))$
        else
            $\mathcal{L}_D = \mathcal{L}_{fm} = f(x, x', z')$
        end if
        $S(x) \Longleftarrow |(1-\alpha)\mathcal{L}_R + \alpha\mathcal{L}_D|$
        end procedure.
    end for
end while
return $S(x)$
```

built based on the data captured by DARPA'98, and it can simulate four attack scenarios well: DoS, probe, U2R and R2L. NSL-KDD is an iterative and updated version of dataset KDDCUP'99, which discards the shortcomings of previous data sets: redundant records, duplicate records and data imbalance, make attack more realistic. UNSW_NB15 is a dataset mixed with real modern normal and modern network traffic comprehensive attack activities, which can best simulate the traffic activities in the real network environment. It includes a wide range of attack scenarios that contains nine different families of attacks like backdoors, DoS, exploits, fuzzers, or worms etc. For each data set, according to the contaminate rate, we constructe a training and a testing set. The former with only normal data and the latter with both normal and attack data.

KDDCUP'99 [1] contains 805050 records with 41 dimensions features, includes three types: inherent features, content features and traffic features. There are some originally discrete data which are inherent features 'protocol_type', 'service', 'flag', 'land', 'logged_in', 'is_host_login', 'is_guest_Login',

[1] http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

which is processed by using dummy coding or one hot coding. We label the "normal" samples as "abnormal" and the 'abnormal' samples as the 'normal' following the dataset setup of [36] due to the detection in this experiment is pure binary classification, and this trick will not affect the identification ability of model. Next, we randomly divide the original dataset (about 500000 samples) into two groups. Then we choose the normal label samples as the training set from one of the two sets to train our model. We do not consider the abnormal samples (remove them). At last, normal label and abnormal label samples are selected as the test set according to the contaminate rate.

NSL-KDD [2] contains 148517 data patterns with 41 features, but there is additional class label in each sample. It represents nine weeks of raw TCP dump data for a local-area network (LAN) simulating a typical U.S. Air Force LAN. We label each class category into five types respectively: DoS, probe, U2R, R2L, and normal. We will conduct additional multi classification experiments on this dataset. In the preprocessing stage, we first combine the training set and test set. Then each classification feature is encoded into one hot vector or dummy vector and is scaled to be in the range [0,1]. The division of training set and test set is similar to that of KDDCUP'99.

UNSW_NB15 [3] is captured by traffic data which contains data packet headers based on the traffic between hosts (i.e., client to server or server to client). It covers the in-depth characteristics of network traffic, which contains 257673 samples with 49 dimensions features. And it is composed of flow features, basic features, content features, time features and additional generation features. In addition, we use Dummy Encoding or One-Hot Encoding to process three main nominal features like protocol types, state types and services. So these discrete features will be transformed into numeric features. The division of training set and testing set is similar to that of KDDCUP'99.

4.2 Simulation experiments

The network anomaly detection problem is for anomalies without prior knowledge where the latent space distributions between the minority abnormal samples and majority normal samples. For this target, we use the additional encoder to model the latent space distribution of data examples. In addition, we assume that all the training data patterns are normal. Moreover, we use spectral normalization as optimizer to train MLP network with hidden layers for the encoder, and discriminator. We use MLP networks with depth 3 and 1 intermediate layer for the discriminator, and use depth 3 and 1 hidden layer for Generator, and encoder. In order to generate better samples, we find a latent space dimension of 32 is the best choose in our study. However, by introducing an encoder, our proposal is expected to improve both the detection precision and the the detection efficiency. Therefore, we compare our method to the work in [11], [13], [40], [37], which all detect anomalies using GAN architecture and additional network that help reconstructs data samples. The detection

[2] http://205.174.165.80/CICDataset/NSL-KDD/
[3] https://research.unsw.edu.au/projects/unsw-nb15-dataset

performance is evaluated using precision, recall and F1 score. The detection efficiency is evaluated using the mean computing time from the beginning of training to the end of test. In addition, we evaluate the effect of combination of gradient penalty term (GP) and spectral normalization (SN). In a nutshell, we call this ablation studies 4.5. We expect that a better detection rate can be achieved when considering a combination of both GP and SN.

4.3 Comparison algorithms

To validate the effectiveness of our own approach, we compare it with some other anomaly detection methods, such as isolated forests (IF) [23], One Class Support Vector Machine (OC-SVM) [50], autoencoder-based model (DAGMM) [33] and some GAN-based models AnoGAN [10] and ALAD [37]. The following is a brief introduction about these methods:

Isolated Forests (IF) is a classical traditional machine learning method, which is generally used for anomaly detection of structured data. Anomalies are defined as those "outliers easy to be isolated", which is also understood as sparse space distribution. Firstly, the randomly selected segmentation values are utilized to construct a tree on the randomly selected features. Then, the anomaly score is defined as the average path length from a specific sample to the root.

One Class Support Vector Machine (OC-SVM) is an unsupervised novelty detection method based on libsvm, that is used to evaluate the high-dimensional distribution by learning the decision boundary around the normal example.

Deep Autoencoding Gaussian Mixture Model (DAGMM) is a method for anomaly detection using a model of the autoencoder. The training algorithm is based on an algorithm that determines the possibility of latent and reconstruction features of samples as a criterion for anomaly detection. And its main idea is to first train an autoencoder to generate both potential spatial features and reconstructed features of a sample. Then we train an evaluation network, which outputs the Gaussian mixture model parameters of low-dimensional potential space for sample modeling.

AnoGAN is the first anomaly detection method based on GAN. This method uses the common basic architecture DCGAN for unsupervised learning of the latent spatial distribution characteristics of normal samples. Then it restores the latent representation of each test sample in the reference stage to obtain the result that determines sample abnormality when it exceeds a certain threshold.

ALAD is a bidirectional GAN-based adversarial learning method of anomaly detection, which captures adversarial learning features for abnormal detection tasks. Then the reconstruction error is used to determine whether the data sample is abnormal or not. Moreover, the model is built on the basis of the cyclic consistency loss in real space and latent space and the stable GAN training. Its performance of anomaly detection achieves SOTA.

Table 1 Binary classification performance on three datasets

Dataset	Model	Precision	Recall	F1 score
KDDCUP'99	IF	0.9216	0.9373	0.9294
	OC-SVM	0.7457	0.8523	0.7954
	DAGMM	0.9297	0.9442	0.9369
	AnoGAN	0.8786	0.8297	0.8865
	ALAD	0.9427	0.9577	0.9501
	MAD-GAN	0.8691	0.9479	0.9000
	FID-GAN	0.8031	0.8031	0.8859
	GANAD	**0.9749**	**0.9761**	**0.9755**
NSL-KDD	IF	0.9217	0.7831	0.8467
	OC-SVM	0.8328	0.5574	0.7158
	DAGMM	0.7440	0.8928	0.8117
	AnoGAN	0.7222	0.8666	0.7879
	ALAD	0.9264	0.9263	0.9263
	MAD-GAN	0.8001	0.8742	0.8355
	FID-GAN	0.9054	0.8873	0.8084
	GANAD	**0.9583**	**0.9580**	**0.9581**
UNSW_NB15	IF	0.9137	0.7681	0.8346
	OC-SVM	0.4543	0.4418	0.4490
	DAGMM	0.8092	0.9110	0.8571
	AnoGAN	0.8283	0.8801	0.8534
	ALAD	0.9190	0.9210	0.9200
	MAD-GAN	0.7675	0.8640	0.8595
	FID-GAN	0.6083	1	0.7518
	GANAD	**0.9482**	**0.9483**	**0.9482**

MAD-GAN is a method which proposed a multivariate anomaly detection with GAN framework to detect attacks using a novel anomaly score called DR-Score. This score exploits both the discriminator and generator networks, which are LSTM-RNN networks, by computing and combining a reconstruction loss to the discrimination loss.

FID-GAN is a novel fog-based, unsupervised intrusion detection method for CPSs using GANs. It is proposed for a fog architecture, which brings computation resources closer to the end nodes and thus contributes to meeting low-latency requirements.

4.4 Results and Discussion

4.4.1 Detection performance

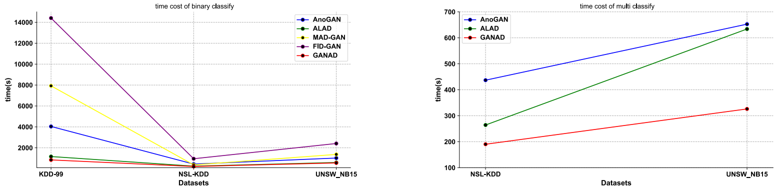
We use the precision, recall and F1 score as the performance metric to evaluate the detection performance of anomalies. Our experimental results of the three datasets are shown in the table 1 2. It demonstrates that our method is better than both GAN-based methods AnoGAN, ALAD, MAD-GAN and FID-GAN by comparing with the above methods. From experimental results, in general, we can observe that:

- For the KDDCUP'99 dataset, refining into each metric, our approach is about 2% and 3% higher than the best baseline methods. In the meantime, it achieves that our approach surpasses other SOTA methods in terms of accuracy. From the experimental results of classify NSL-KDD and UNSW_NB15 datasets in table 1, which are more complex and more challenging to detect intrusions from, since their precision are, in general, lower than the results of the KDDCUP'99 data set.
- For the NSL-KDD dataset, our approach significantly outperforms all the above methods in all metrics. It is also known from the result that the lack of sufficient training data will leads to the poor performance of all the data-based deep learning methods. We guess that MAD-GAN and FID-GAN perform not well since NSL-KDD dataset is more complex and more challenging. Given that our method can accurately learn the distribution of real normal and generated data, and identify the subtle differences between the data. Hence it is good for efficient detection even if the dataset is not large.
- In UNSW_NB15 dataset, our method is slightly better than ALAD and AnoGAN, but it outperforms the other methods in terms of accuracy and F1 scores. In the precision case, IF is higher than AnoGAN. However, the other two cases of IF results are weaker than the latter deep learning methods, which again verify the deficiency of machine learning methods in discriminating anomalies. FID-GAN's precison seems poor, but it achieves a near 100% recall value. This is unacceptable in the real-world setting as the number for false positive sample is too large.

We oberseve from the result that MAD-GAN not considering the computing complexity would hinder the intrusion detection performance. FID-GAN is the improved version of MAD-GAN which is still Inefficient. In addition, FID-GAN adopting computing optimization is not suitable to discriminate anomalies since gradient-based optimizer is easy to get struck in local optimal. Because of the disadvantage of the KDDCUP'99 dataset itself which is no classification or specification of specific attack categories, it is not suitable for the multi classification task. Of note, MAD-GAN and FID-GAN are not able to be employed to the specific category discrimination. However,it still can be observed from table 2 that our approach completely surpasses various existing baseline approaches. And the results show excellent detection performance of our approach. This is because novel training strategy, which is used in our method, is capable of learning more complex data distributions better than other GAN-based method. Overall, looking at the relative performance of GANAD with other GAN-based methods, we can see that GAN-based anomaly detection is unable compete our method since we model latent space distribution of samples appropriately.

Table 2 Multi classification performance on two datasets

Dataset	NSL-KDD		UNSW_NB15	
Metrics	Precison	Accuracy	Precison	Accuracy
IF	0.2533	0.7445	0.0232	0.0232
One-Class SVM	0.2854	0.7146	0.4564	0.4418
DAGMM	0.8666	0.8714	0.9010	0.9009
AnoGAN	0.8775	0.8837	0.9021	0.9024
ALAD	0.8154	0.8238	0.9021	0.9022
GANAD	**0.9082**	**0.9125**	**0.9802**	**0.9803**



(a) The time cost of binary classification experiments

(b) The time cost of multi classification experiments

Fig. 6 Time cost comparison between GANAD and the two other methods

4.4.2 Time cost performance

To validate the efficiency of our approach, we compare the time spent on experiments to other GAN-based methods. We train the model for 50 epochs for all methods. From the results of figure 6(a) and figure 6(b), it is obvious that our methodology is better than the other two methods. From experimental results, in general, we can observe that:

- Since the detection of anomalies is a latency constrained application, the anomaly detection score needs to be computed in a short time. This mainly depends on the computation of the discrimination and reconstruction losses. So, most GAN-based methods are suffer from time consuming. In the KDDCUP'99 dataset, the results in figure 6(a) show that our approach is significantly superior to AnoGAN. In contrast to our architecture, MAD-GAN and FID-GAN model data as time series and use RNN-LSTM networks to consider data dependencies. Thus, our method only using fully connected layers compared with them requires a lower computing time. It also indicates that we can deal with big high-dimension data faster and efficiently. In addition, from the time cost in NSL-KDD and UNSW_NB15 datasets, we can see the time difference between our method and the other four methods are not so obvious. We guess the reason is that the datasets is not big enough for experiment. Even so, our evaluation results are still the best.
- For multi classification experimental results, figure 6(b) shows the advantage of our approach over the other two methods that GANAD is good

Table 3 Binary classification performance of ablation study on three datasets

Model	Precision	Recall	F1 score
KDDCUP'99			
Basemodel	0.9733	0.9761	0.9745
Basemodel+GP	0.9730	0.9761	0.9745
Basemodel+SN	0.9735	0.9761	0.9748
Basemodel+GP+SN	**0.9749**	0.9761	**0.9755**
NSL-KDD			
Basemodel	0.9493	0.9492	0.9493
Basemodel+GP	0.9542	0.9541	0.9541
Basemodel+SN	0.9576	0.9575	0.9575
Basemodel+GP+SN	**0.9583**	**0.9580**	**0.9581**
UNSW_NB15			
Basemodel	0.9188	0.9194	0.9191
Basemodel+GP	0.9189	0.9188	0.9189
Basemodel+SN	0.9188	0.9191	0.9190
Basemodel+GP+SN	**0.9482**	**0.9483**	**0.9482**

at multi-classification. The cost time of ALAD is about twice that of our approach, which validates the success of the combination of gradient penalty and spectral normalization. This is because finding the latent representation of a sample and computing its reconstruction loss demands time. And the encoder in our architecture enables a major reduction in the time taken to detect anomalies because it obtains the latent representation of patterns through a direct mapping. Since training GANs is not always an easy task due to mode collapse and stabilization issues, this is a disadvantage in the use of ALAD for improving existing GAN-based IDSs. In contrary to ALAD, our method stabilize the training by adding constrain to loss computation.

4.5 Ablation studies

To better demonstrate the detection performance of our model, we perform ablation experiments by adding and deleting model components. In particular, we perform experiments with and without gradient penalty term (GP) optimization, with spectral normalization and without spectral normalization (SN) to examine the performance of the full model (with gradient penalty term and spectral normalization) respectively. From experimental results, in general, we can observe that:

- As shown in table 3, generally speaking, our approach is overall balanced in all aspects of metrics, and the addition of both SN and GP can improve the model performance on the UNSW_NB15 dataset. But the addition of GP and SN alone does not seem to work more significantly on the KDDCUP'99 and NSL-KDD datasets.

Table 4 Multi classification performance of ablation study on two datasets

Dataset	NSL-KDD		UNSW_NB15	
Metrics	Precison	Accuracy	Precison	Accuracy
Basemodel	0.9065	0.9101	0.9802	0.9799
Basemodel+GP	0.9009	0.9013	0.9800	0.9800
Basemodel+SN	0.9065	0.9101	0.9799	0.9790
Basemodel+GP+SN	**0.9082**	**0.9125**	**0.9802**	**0.9803**

- From the table 4, we can see that our ablation experiments do not well reflect the superiority of the overall framework well, and the performance of each variant approach is almost the same. In the NSL-KDD experiment, our variant approaches have been slightly improved, which indicates that the effectiveness of the experiment still exists. In the UNSW_NB15 dataset, either adding gradient penalty or adding spectral normalization can only make the model more stable or more computational efficient. We guess the reason is related to the large difference in the number of attack types.

5 Conclusion and future work

In this artical, we proposed GANAD, a novel system using a GAN which is specifically-designed for detecting network anomalies. The detection is based on the novel training strategy, which can better learn minority abnormal distribution from normal data patterns. In addition, we utilize a additional encoder to mapping data samples to the latent space, such that the generator loss computation is optimized. Furthermore, to address the severe GAN unstable training problem that hinders the detection task, our approach is proposed within discriminator training replace JS divergence with Wasserstein distance adding gradient penalty. The empirical evaluation on three datasets demonstrates that our model outperforms the previous GAN-based model in most cases with respect to recall, precision, F1 score. In addition, it reduces the training cost and time consumption. Moreover, we further conduct ablation experiments to validate the effectiveness of our method. Therefore, our approach provides a new way to detect network anomalies.

The information about the distribution of anomalous samples in the existing network data is so vague and undetectable, hence abnormal behavior are only slightly deviated and masked in the data space. In future works, we plan to explore network anomaly detection at a deeper level. we will investigate the use of GANs in the unsupervised detection of cyber-intrusion and approaches to further enhance the detection performance of unknown abnormal traffic.

Acknowledgments. This work was supported by the National Key Research and Development Program of China (No.2020YFB1805400) and the National Natural Science Foundation of China (61876134).

Declarations

- Ethical Approval
  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.
- Competing interests
  The authors declare that there are no competing interest regarding the publication of this article.
- Authors' contributions
  Jie Fu wrote the main manuscript text, Jianpeng Ke and Kang Yang investigated the manuscript. Lina Wang and Rongwei Yu supervise the manuscript. All authors reviewed the manuscript.
- Funding
  This work was supported by the National Key Research and Development Program of China (No.2020YFB1805400) and the National Natural Science Foundation of China (61876134).
- Availability of data and materials
  The data used to support the findings of this study are available from the corresponding author upon request.

References

[1] Lin, P., Ye, K., Xu, C.-Z.: Dynamic network anomaly detection system by using deep learning techniques. In: International Conference on Cloud Computing, pp. 161–176 (2019). Springer

[2] Chou, D., Jiang, M.: A survey on data-driven network intrusion detection. ACM Computing Surveys (CSUR) **54**(9), 1–36 (2021)

[3] Ahmim, A., Maglaras, L., Ferrag, M.A., Derdour, M., Janicke, H.: A novel hierarchical intrusion detection system based on decision tree and rules-based models. In: 2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS), pp. 228–233 (2019). IEEE

[4] Miao, X., Liu, Y., Zhao, H., Li, C.: Distributed online one-class support vector machine for anomaly detection over networks. IEEE transactions on cybernetics **49**(4), 1475–1488 (2018)

[5] Pang, G., Cao, L., Chen, L., Liu, H.: Learning representations of ultrahigh-dimensional data for random distance-based outlier detection. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2041–2050 (2018)

[6] Pang, G., Shen, C., Jin, H., Hengel, A.v.d.: Deep weakly-supervised anomaly detection. arXiv preprint arXiv:1910.13601 (2019)

[7] Ruff, L., Vandermeulen, R.A., Görnitz, N., Binder, A., Müller, E., Müller, K.-R., Kloft, M.: Deep semi-supervised anomaly detection. In: International Conference on Learning Representations (2019)

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

[9] Gill, P., Jain, N., Nagappan, N.: Understanding network failures in data centers: measurement, analysis, and implications. In: Proceedings of the ACM SIGCOMM 2011 Conference, pp. 350–361 (2011)

[10] Schlegl, T., Seeböck, P., Waldstein, S.M., Schmidt-Erfurth, U., Langs, G.: Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In: International Conference on Information Processing in Medical Imaging, pp. 146–157 (2017). Springer

[11] Akcay, S., Atapour-Abarghouei, A., Breckon, T.P.: Ganomaly: Semi-supervised anomaly detection via adversarial training. In: Asian Conference on Computer Vision, pp. 622–637 (2018). Springer

[12] Pang, G., Shen, C., Cao, L., Hengel, A.V.D.: Deep learning for anomaly detection: A review. ACM Computing Surveys (CSUR) **54**(2), 1–38 (2021)

[13] Li, D., Chen, D., Jin, B., Shi, L., Goh, J., Ng, S.-K.: Mad-gan: Multivariate anomaly detection for time series data with generative adversarial networks. In: International Conference on Artificial Neural Networks, pp. 703–716 (2019). Springer

[14] Donahue, J., Krähenbühl, P., Darrell, T.: Adversarial feature learning. arXiv preprint arXiv:1605.09782 (2016)

[15] Xiong, L., Póczos, B., Schneider, J.: Group anomaly detection using flexible genre models. Advances in neural information processing systems **24** (2011)

[16] Blowers, M., Williams, J.: Machine learning applied to cyber operations. In: Network Science and Cybersecurity, pp. 155–175. Springer, ??? (2014)

[17] Khan, M.S.A.: Rule based network intrusion detection using genetic algorithm. International Journal of Computer Applications **18**(8), 26–29 (2011)

[18] Shone, N., Ngoc, T.N., Phai, V.D., Shi, Q.: A deep learning approach to network intrusion detection. IEEE transactions on emerging topics in computational intelligence **2**(1), 41–50 (2018)

[19] Mulay, S.A., Devale, P., Garje, G.: Intrusion detection system using support vector machine and decision tree. International journal of computer applications **3**(3), 40–43 (2010)

[20] Selvakumar, K., Karuppiah, M., SaiRamesh, L., Islam, S.H., Hassan, M.M., Fortino, G., Choo, K.-K.R.: Intelligent temporal classification and fuzzy rough set-based feature selection algorithm for intrusion detection system in wsns. Information Sciences **497**, 77–90 (2019)

[21] Breunig, M.M., Kriegel, H.-P., Ng, R.T., Sander, J.: Lof: identifying density-based local outliers. In: Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, pp. 93–104 (2000)

[22] Peña, D., Prieto, F.J.: Multivariate outlier detection and robust covariance matrix estimation. Technometrics **43**(3), 286–310 (2001)

[23] Liu, F.T., Ting, K.M., Zhou, Z.-H.: Isolation forest. In: 2008 Eighth Ieee International Conference on Data Mining, pp. 413–422 (2008). IEEE

[24] Torres, P., Catania, C., Garcia, S., Garino, C.G.: An analysis of recurrent neural networks for botnet detection behavior. In: 2016 IEEE Biennial Congress of Argentina (ARGENCON), pp. 1–6 (2016). IEEE

[25] Deng, A., Hooi, B.: Graph neural network-based anomaly detection in multivariate time series. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 4027–4035 (2021)

[26] Kwon, D., Natarajan, K., Suh, S.C., Kim, H., Kim, J.: An empirical study on network anomaly detection using convolutional neural networks. In: ICDCS, pp. 1595–1598 (2018)

[27] Zhao, G., Zhang, C., Zheng, L.: Intrusion detection using deep belief network and probabilistic neural network. In: 2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC), vol. 1, pp. 639–642 (2017). IEEE

[28] Pang, G., van den Hengel, A., Shen, C., Cao, L.: Toward deep supervised anomaly detection: Reinforcement learning from partially labeled anomaly data. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, pp. 1298–1308 (2021)

[29] Wang, H., Pang, G., Shen, C., Ma, C.: Unsupervised representation learning by predicting random distances. In: Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence, pp. 2950–2956 (2021)

[30] Pang, G., Shen, C., van den Hengel, A.: Deep anomaly detection with deviation networks. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 353–362 (2019)

[31] Zhou, C., Paffenroth, R.C.: Anomaly detection with robust deep autoencoders. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 665–674 (2017)

[32] An, J., Cho, S.: Variational autoencoder based anomaly detection using reconstruction probability. Special Lecture on IE **2**(1), 1–18 (2015)

[33] Zong, B., Song, Q., Min, M.R., Cheng, W., Lumezanu, C., Cho, D., Chen, H.: Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In: International Conference on Learning Representations (2018)

[34] Zhai, S., Cheng, Y., Lu, W., Zhang, Z.: Deep structured energy based models for anomaly detection. In: International Conference on Machine Learning, pp. 1100–1109 (2016). PMLR

[35] Mirsky, Y., Doitshman, T., Elovici, Y., Shabtai, A.: Kitsune: An ensemble of autoencoders for online network intrusion detection. In: Network and Distributed Systems Security (NDSS) Symposium (2018)

[36] Zenati, H., Foo, C.S., Lecouat, B., Manek, G., Chandrasekhar, V.R.: Efficient gan-based anomaly detection. arXiv preprint arXiv:1802.06222 (2018)

[37] Zenati, H., Romain, M., Foo, C.-S., Lecouat, B., Chandrasekhar, V.: Adversarially learned anomaly detection. In: 2018 IEEE International Conference on Data Mining (ICDM), pp. 727–736 (2018). IEEE

[38] Mohammadi, B., Sabokrou, M.: End-to-end adversarial learning for intrusion detection in computer networks. In: 2019 IEEE 44th Conference on Local Computer Networks (LCN), pp. 270–273 (2019). IEEE

[39] Schlegl, T., Seeböck, P., Waldstein, S.M., Langs, G., Schmidt-Erfurth, U.: f-anogan: Fast unsupervised anomaly detection with generative adversarial networks. Medical image analysis **54**, 30–44 (2019)

[40] de Araujo-Filho, P.F., Kaddoum, G., Campelo, D.R., Santos, A.G., Macêdo, D., Zanchettin, C.: Intrusion detection for cyber–physical systems using generative adversarial networks in fog environment. IEEE Internet of Things Journal **8**(8), 6247–6256 (2020)

[41] Huang, S., Lei, K.: Igan-ids: An imbalanced generative adversarial network towards intrusion detection system in ad-hoc networks. Ad Hoc Networks **105**, 102177 (2020)

[42] Yuan, D., Ota, K., Dong, M., Zhu, X., Wu, T., Zhang, L., Ma, J.: Intrusion detection for smart home security based on data augmentation with edge computing. In: ICC 2020-2020 IEEE International Conference on Communications (ICC), pp. 1–6 (2020). IEEE

[43] Flores, S.: Variational Autoencoders Are Beautiful. https://www.compthree.com/blog/autoencoder/ (2019)

[44] Miyato, T., Kataoka, T., Koyama, M., Yoshida, Y.: Spectral normalization for generative adversarial networks. arXiv preprint arXiv:1802.05957 (2018)

[45] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C.: Improved training of wasserstein gans. Advances in neural information processing systems **30** (2017)

[46] Roth, K., Lucchi, A., Nowozin, S., Hofmann, T.: Stabilizing training of generative adversarial networks through regularization. Advances in neural information processing systems **30** (2017)

[47] Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein generative adversarial networks. In: International Conference on Machine Learning, pp. 214–223 (2017). PMLR

[48] Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.A.: A detailed analysis of the kdd cup 99 data set. In: 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, pp. 1–6 (2009). Ieee

[49] Moustafa, N., Slay, J.: Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set). In: 2015 Military Communications and Information Systems Conference (MilCIS), pp. 1–6 (2015). IEEE

[50] Schölkopf, B., Williamson, R.C., Smola, A., Shawe-Taylor, J., Platt, J.: Support vector method for novelty detection. Advances in neural information processing systems **12** (1999)