Ad Hoc Networks 105 (2020) 102177

Contents lists available at ScienceDirect

Ad Hoc Networks

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

IGAN-IDS: An imbalanced generative adversarial network towards intrusion detection system in ad-hoc networks

Shuokang Huang[a], Kai Lei[a,b,*]

[a] ICNLAB, School of Electronic and Computer Engineering (SECE), Peking University, Shenzhen 518055, PR China
[b] PCL Research Center of Networks and Communications, Peng Cheng Laboratory, Shenzhen, China

ARTICLE INFO

Article history:
Received 3 December 2019
Revised 26 February 2020
Accepted 12 April 2020
Available online 25 April 2020

Keywords:
Intrusion detection system
Class imbalance
Generative adversarial network
Ad-hoc networks

ABSTRACT

With the emergence of ever-advancing network threats, the guarantee of system security becomes increasingly crucial, especially in the dynamic and decentralized ad-hoc networks. One essential part of cybersecurity is intrusion detection, which identifies anomalous activities according to traffic patterns. However, the class-imbalanced data have caused a challenging problem where the number of abnormal samples is significantly lower than that of the normal ones. This class imbalance problem confines the performance of intrusion classifiers and results in low robustness to unknown anomalies. In this paper, we propose a novel Imbalanced Generative Adversarial Network (IGAN) to tackle the class imbalance problem. In the primary novelty of our model, we introduce an imbalanced data filter and convolutional layers to the typical GAN, generating new representative instances for minority classes. Further, an IGAN-based Intrusion Detection System, namely IGAN-IDS, is established to cope with class-imbalanced intrusion detection, using the instances generated by IGAN. Concretely, IGAN-IDS consists of three modules: feature extraction, IGAN, and deep neural network. First, we utilize a feed-forward neural network (FNN) to transform raw network attributes into feature vectors. Then, the IGAN generates new samples expressed in the latent space. Finally, the deep neural network, composed of convolutional layers and fully-connected layers, executes the final intrusion detection. We conduct experiments on three benchmark datasets to evaluate the performance of IGAN-IDS, comparing against 15 other methods. The experimental results demonstrate that our proposed IGAN-IDS outperforms the state-of-the-art approaches.

1. Introduction

Network security has emerged as one of the major issues in computer systems, especially in the dynamic and decentralized ad-hoc networks. Ad-hoc networks are self-organized with no fixed security scheme, leading to a higher risk of arising security problems [1]. Though there are massive mechanisms to enhance the defense of network [2], the existing methods remain inadequate for security guarantee against the ever-changing attacks. For example, in May 2017, the ransomware WannaCry attacked numerous devices over 150 countries, having more than 300,000 victims [3].

Generally, network intrusions involve misuse behaviors driven by known attacks (e.g., denial-of-service, worm) and anomalous activities rooted in unknown threats (e.g., botnet, malware attack) [4]. Misuse behaviors can be detected based on malicious traffic patterns, while anomaly detection identifies unknown attacks via learning from the historical data [5]. Therefore, it's necessary for intrusion detection to efficiently recognize known attacks and realize high robustness on detecting unknown anomalies, simultaneously.

Confronted by the above threats, the Intrusion Detection System (IDS) is the essential part of network security [6]. IDS aims to automatically identify whether there is a malicious activity or policy violation by detecting anomalous network patterns. Most prior works implement intrusion detection with conventional statistical learning methods, such as Naive Bayes [7], Decision Tree [8], Random Forest [9] and Support Vector Machine [10–14]. Inspired by the remarkable effect of deep learning, several recent studies employed neural networks for intrusion detection, including Multilayer Percetron [15], Convolutional Neural Network [16] and Recurrent Neural Network [17]. Further, Aljawarneh et al. [18] introduced an intrusion detection system based on feature selection and hybrid algorithm, realizing an improved accuracy.

Though the previous approaches have made profound progress, the class-imbalanced data are still a challenging problem that hinders the performance of most intrusion detection systems [19]. The class imbalance problem occurs when the number of intrusion samples is significantly lower than that of the normal ones.

* Corresponding author
E-mail addresses: huangshk6@pku.edu.cn (S. Huang), leik@pkusz.edu.cn (K. Lei).

https://doi.org/10.1016/j.adhoc.2020.102177
1570-8705/

This problem is very prevalent in intrusion detection since normal activities are dominating in real networks. Quantitatively, the imbalance ratio between the majority class and the minority classes is a metric to estimate the degree of class imbalance. For instance, CICIDS2017 [20] is a real dataset of network intrusion, containing 2,271,320 normal activities and only 556,509 intrusion samples, where the imbalance ratio is 4.0814: 1. The challenge is that most methods are in favor of the majority class while neglecting the minority classes, as the samples of minority classes are too few and inadequate for those methods. The models cannot learn enough information about the minority classes from limited samples and result in a bias toward the majority class. However, the performance of detecting intrusion samples (minority) is the emphasis. The misdetection of an intrusion activity is much more harmful than classifying the normal one as an intrusion.

To address the class imbalance problem in intrusion detection, we propose a novel Imbalance Generative Adversarial Network (IGAN) to generating representative samples for minority classes. IGAN filters the data to ensure only generating samples for minority classes, aiming at more precise intrusion detection. An IGAN-based Intrusion Detection System (IGAN-IDS) is built to cope with class imbalance intrusion detection and includes three modules: feature extraction, IGAN, and deep neural network. The key contributions of this paper are listed as follows.

(1) This work proposes a novel IGAN to generate representative samples for minority classes, countering the class imbalance problem in intrusion detection. The typical Generative Adversarial Network (GAN) aims at generating samples without regarding their classes [21], and the usage of Multilayer Perceptron limits its expression ability. Therefore, we introduce a filter in IGAN for resampling the minority classes. We further consider the classes as conditions to train IGAN [22]. Moreover, we join convolutional layers in the Generator G, enhancing its expression ability.
(2) Based on IGAN, we present IGAN-IDS, which execute intrusion detection more accurately. To the best of our knowledge, it's the first work to apply the GAN-based method to increase the performance of intrusion detection. And different from the previous methods, we establish IGAN-IDS to solve multiclass classification problems, which reflects the practicability and flexibility of IGAN-IDS.
(3) We conduct comparative experiments on three intrusion detection datasets, illustrating that IGAN-IDS outperforms the state-of-the-art methods. Furthermore, we discuss the robustness of IGAN-IDS with different generated ratios and different imbalance ratios, showing the robust performance of IGAN-IDS under different circumstances. Finally, we perform an ablation study to evaluate the effectiveness of each module in IGAN-IDS.

The rest of this paper is organized as follows. In Section 2, we outline multiple state-of-the-art IDS models and give an overview of class balancing methods, including GAN. Section 3 introduces our proposed IGAN model and the detail of IGAN-IDS, where we integrate the IGAN module with a feature extraction module and a deep neural network module. Section 4 presents the experimental details of comparative study, robustness study, and ablation study. Finally, we draw a conclusion and highlight our future works in Section 5.

2. Related work

2.1. Intrusion detection system (IDS)

Intrusion Detection System plays an essential role in various information systems, including ad-hoc networks. It enhances the security of cyber systems by detecting anomalies in multiple means.

Some researches [23,24] have introduced methods based on existing technologies to conduct intrusion detection. Panda et al. [7] discussed the use of Naive Bayes in network intrusion detection, while [8] compared Naive Bayes based IDS to Decision Tree. Zhang et al. [9] pioneered the practice of Random Forest (RF) for automatic intrusion detection, and show superior capabilities than the previous models. Additionally, Support Vector Machine (SVM) was widely used in IDS [10–12] and realized high accuracy detection. Using Least Square SVM (LSSVM), Amiri et al. [13] performed Mutual Information based Feature Selection (MIFS) to increase the detection efficiency. Ambusaidi et al. [14] proposed Flexible MIFS (FMIFS) and adopt LSSVM for intrusion detection, but advanced in time complexity. However, these conventional approaches are limited by their inadequate expression ability on data and have difficulties in dealing with imbalanced data.

Along with the rise of deep learning, multiple varieties of neural networks [4,5] are employed for intrusion detection, bringing better performance. Based on Multilayer Perceptron (MLP), Moradi et al. [15] built neural networks with different amounts of layers, showing the applicability of deep learning in intrusion detection. Li et al. [16] applied a Convolutional Neural Network (CNN) for representation learning in intrusion detection and illustrated its superiority against the traditional models. Besides, Chawla et al. [17] combined CNN with Recurrent Neural Network (RNN), where CNN captures the local features, and RNN learns sequential correlations. Using a Feed-forward Neural Network, Ashfaq et al. [25] performed a fuzziness categorization on the samples to further improve the generalization of IDS. Based on a random neural network and the artificial bee colony algorithm, Qureshi et al. [26] brought forward a new intrusion detection system, leading to superiority over the traditional approaches. To compare the effectiveness of different algorithms, Ahmad et al. [27] adopted SVM, RF, and extreme learning machine (ELM) for intrusion detection, respectively, and the results indicated that ELM achieved the best performance.

Although the adoption of neural networks strengthens the expression ability of data, an unavoidable drawback is that the class imbalance problem remains, and these models cannot foresee unknown anomalies. In our proposed IDS, we introduced an IGAN module to balance the classes, and the synthesized samples from IGAN can simulate the emergence of unknown anomalies.

2.2. Class balancing methods

Class imbalance problem is a long-term challenge in the machine learning area [28], which drives difficulties for intrusion detection as well. This problem happens whenever there is a disproportion in the number of samples among multiple classes. Most algorithms are sensitive to the imbalance problem. When this problem occurs, the majority class may overwhelm the algorithms, while the minority classes may be neglected, as the algorithms aim at high overall accuracy on all samples [29].

Common balancing techniques include random under-sampling (RUS) and random over-sampling (ROS) [30]. Using RUS, samples of the majority class will be randomly removed. On the other hand, ROS suggests randomly duplicating samples in the minority classes. In the research area of intrusion detection, the class imbalance problem has attracted attention from researchers [31,32] as well. Under-sampling and over-sampling algorithms were designed to counter this problem in IDS [33,34]. Cieslak et al. [35] further implemented a combination of under-sampling and over-sampling to combat the imbalance. But, under-sampling may cause a loss of useful information, while over-sampling replicates the samples and leads to possibly overfitting.

It is worth noting that the Synthetic Minority Over-sampling Technique (SMOTE) [36,37] provided petty good performance in many models [38], including IDS models [39,40]. However, SMOTE

depended on interpolation to conduct over-sampling, which results in low representative synthesized samples. The synthesized samples of SMOTE may overlap at existing data when the distances between different classes are too close, which even makes the classification result worse.

Therefore, to deal with the class imbalance problem in intrusion detection, we adopt an IGAN module to generate more representative samples for minority classes.

2.3. Generative adversarial network

Generative Adversarial Network (GAN) [21] is a framework to learn from unknown data distribution and generate similar samples. GAN introduces two models, a generative model G, and a discriminative model D. G implicitly draws a generative distribution of new samples, while D distinguishes them from the real ones. After a minimax two-player gaming between these models, the generative distribution of G can express the real one, while D cannot distinguish the two distributions and converges to 0.5.

To optimize GAN, it's vital to properly measure the distance between the generative distribution p_g and the real distribution p_{data} [21]. The most commonly used mean to quantify the distance is f-divergence $D_f(p_{data}\|p_g)$ with a convex function f as follows [41]:

$$D_f(p_{data} \parallel p_g) = \int_x p_g(x) f\left(\frac{p_{data}(x)}{p_g(x)}\right) dx, \tag{1}$$

where $f(1) = 0$ indicates that p_g and p_{data} are equivalent. In the origin of GAN, the Jensen-Shannon (JS) divergence [42] was adopted as f:

$$f(t) = t \log(t) - (t+1)\log(t+1). \tag{2}$$

Generally, the optimization of GAN is minimizing the f-divergence, which will be described thoroughly in Section 3.

Due to its capability to represent complex data, GAN has been widely applied in diverse domains [41], including computer vision [43], natural language processing [44], dynamic networks [45] and security [46]. The high-dimensional expression ability of GAN contributes to counter the class imbalance problem as well. For example, Douzas and Bacao [47] used a conditional GAN [22] and resulted in an improvement of generating minority class data. Vu et al. [48] conducted network traffic classification with Auxiliary Classifier GAN (AC-GAN) to handle the imbalance problem.

In this paper, we develop an Imbalanced GAN (IGAN) to cope with the class imbalance problem in intrusion detection, via generating samples for minority classes adaptively. With the synthesized samples, the IGAN-based IDS (IGAN-IDS) can achieve better performance. To be more specific, IGAN-IDS firstly turns network attributes into latent feature vectors and learn from the distribution. Later, IGAN-IDS utilizes synthesized feature vectors to enhance intrusion detection.

3. Methodology

In this section, we propose an Imbalanced Generative Adversarial Network (IGAN) model to generate new samples for the minority classes. And we build an IGAN-based Intrusion Detection System (IGAN-IDS) to perform class imbalance intrusion detection.

3.1. Imbalanced generative adversarial network (IGAN)

We strengthen the typical Generative Adversarial Network (GAN) [21] to alleviate the class imbalance problem in intrusion detection by generating samples for minority classes.

A typical GAN includes a generative model (Generator, G) and a discriminative model (Discriminator, D). G takes noises $\boldsymbol{z}$ as inputs and reconstructs them into synthesized samples $G(\boldsymbol{z})$, which implicitly defines a generative distribution $p_g(\boldsymbol{z})$. D takes sample $\boldsymbol{x}$ as input and outputs the probability $D(\boldsymbol{x})$ that $\boldsymbol{x}$ comes from the real distribution $p_{data}(\boldsymbol{x})$ rather than $p_g(\boldsymbol{x})$. In other words, G maps $\boldsymbol{z}$ from a noise distribution $p_z(\boldsymbol{z})$ to $p_g(\boldsymbol{z})$, while D distinguishes $p_{data}(\boldsymbol{x})$ from $p_g(\boldsymbol{x})$. G is to maximize the probability that D makes a mistake. The optimization of GAN is performed to increase the similarity between $p_{data}(\boldsymbol{x})$ and $p_g(\boldsymbol{z})$. Originally, Jensen-Shannon (JS) divergence [42] is used to measure this similarity as follows:

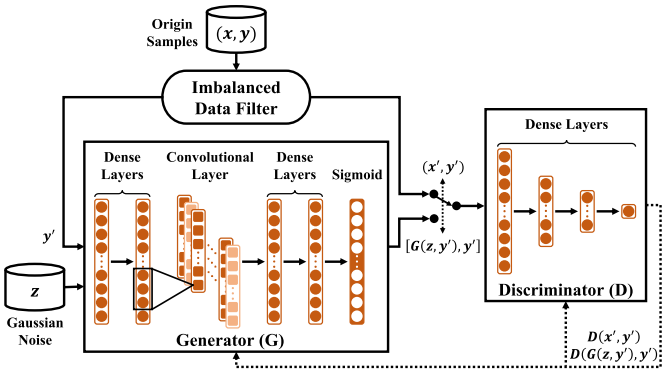

Fig. 1. Model architecture of IGAN.

$$\begin{aligned} &JS(p_{data} \parallel p_g) = \tfrac{1}{2}KL(p_{data} \parallel p_m) + \tfrac{1}{2}KL(p_g \parallel p_m), \\ &p_m = \tfrac{1}{2}(p_{data} + p_g), \end{aligned} \tag{3}$$

where KL is the Kullback-Leibler divergence [49]. Hence, G minimize $JS(p_{data}\|p_g)$, while D maximize $JS(p_{data}\|p_g)$, which can be formulated as:

$$\begin{aligned} \min_G \max_D V(D,G) = \min_G \max_D \Big(&\mathbb{E}_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})}[\log D(\boldsymbol{x})] \\ &+ \mathbb{E}_{\boldsymbol{z}\sim p_z(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))] \Big), \end{aligned} \tag{4}$$

where $V(D, G)$, the concrete expression of $JS(p_{data}\|p_g)$, denotes the value function of GAN. After training G and D in alternation, GAN reaches an equilibrium where $p_g = p_{data}$, and D converges to 0.5 as:

$$D(\boldsymbol{x}) = \frac{p_{data}(\boldsymbol{x})}{p_{data}(\boldsymbol{x}) + p_g(\boldsymbol{x})}. \tag{5}$$

However, the typical GAN aims at generating samples without regarding their classes, intrinsically not coping with class imbalance. And the typical GAN employs a Multilayer Perceptron in G, leading to weak expression ability. Therefore, we utilize an imbalanced data filter in IGAN for the minority classes, which determines the generating quantity as well. We further consider the classes as conditions to train IGAN, together with the network attributes. Moreover, we reform the structure of model G by joining convolutional layers, enhancing the expression ability.

3.1.1. Model architecture

Depicted in Fig. 1, IGAN consists of three components: imbalanced data filter, Generator G, and Discriminator D. The imbalanced data filter samples from minority classes and calculates the generating quantity. After that, G and D conduct adversarial learning. In summary, IGAN takes original samples $\boldsymbol{s} = (\boldsymbol{x}, \boldsymbol{y})$ as inputs, and outputs synthesized samples $\boldsymbol{s}_G = [G(\boldsymbol{z}, \boldsymbol{y}'), \boldsymbol{y}']$, where $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{z}$ and $\boldsymbol{y}'$ denotes original feature vectors, original class labels, noises and minority class labels, respectively.

a) Imbalanced data filter: As shown in Fig. 1, the filter takes original sample $\boldsymbol{s} = (\boldsymbol{x}, \boldsymbol{y})$ as input, and outputs samples $\boldsymbol{s}' = (\boldsymbol{x}', \boldsymbol{y}')$ of minority classes. In general, the procedure of filter can be formu-

lated as:

$$\boldsymbol{s'} = \left\{ s' = \left(x', y'\right) \;\middle|\; s' \in \boldsymbol{s}, y' \neq \underset{c_\tau \in \boldsymbol{c}}{\operatorname{argmax}}(n_{c_\tau}) \right\}, \tag{6}$$

where $\boldsymbol{c} = \{c_1, \ldots, c_\tau\}$ is the set of different classes and n_{c_τ} indicates the number of samples in class c_τ.

We set a generated ratio (denoted as r) to quantitatively decrease the imbalance ratio, which indicates the proportion of the synthesized samples and the real samples. It's formed as $r = a : b$ where a and b is the number of synthesized samples and real samples, respectively. Given a generated ratio, which is set to 1: 1 by default, the filter further calculates the generating quantity $\boldsymbol{k}$ for each class, as follows:

$$\boldsymbol{k} = \left\{ k_{c_\tau} = n_{c_\tau} \times r \;\middle|\; c_\tau \in \boldsymbol{c} \right\}, \tag{7}$$

where k_{c_τ} is the generating quantity of class c_τ.

Finally, $\boldsymbol{s'}$ are applied by G and D for adversarial learning, and $\boldsymbol{k}$ samples will be generated by G to counter the class imbalance.

b) Discriminator D: As Fig. 1, the discriminative model D is composed of a Multilayer Perceptron. In adversarial learning, D alternatively takes G's outputs $G(\boldsymbol{z}, \boldsymbol{y'})$ or the ground-truth feature vectors $\boldsymbol{x'}$, along with the corresponding class labels $\boldsymbol{y'}$, as input. It's worth noting that $\boldsymbol{y'}$ should be embedded in one-hot vectors before the concatenation. For a specific input $(\dot{\boldsymbol{x}}, \dot{\boldsymbol{y}})$, D estimates the probability $D(\dot{\boldsymbol{x}}, \dot{\boldsymbol{y}})$ that it comes from the real samples rather than from $\boldsymbol{s}_G$.

The fully-connected layers $\boldsymbol{d}$ can be formalized as follows:

$$\boldsymbol{d} = max(0, \boldsymbol{\omega_d}\ \boldsymbol{v} + \boldsymbol{b_d}), \tag{8}$$

where $\boldsymbol{v}$ is the input of each layer, while $\boldsymbol{\omega_d}$ and $\boldsymbol{b_d}$ is the weights and bias respectively. Moreover, we design the output layer of D to be linear, meaning that D directly outputs the result of the aftermost fully-connected layer.

c) Generator G: As Fig. 1, we implement G with multiple fully-connected layers, convolutional layers, and a sigmoid output layer. G takes minority class labels $\boldsymbol{y'}$ and Gaussian noises $\boldsymbol{z}$ as input and generates feature vectors $\boldsymbol{x}_G = G(\boldsymbol{z}, \boldsymbol{y'})$ for minority classes. Concretely, we integrate $\boldsymbol{z}$ and $\boldsymbol{y'}$ by concatenating them to $\boldsymbol{v} = [\boldsymbol{z}; \boldsymbol{y'}]$ [22]. $\boldsymbol{y'}$ should be embedded in one-hot vectors before the concatenation, which is the same as D. The formalization of fully-connected layers can refer to (8).

The convolutional layer executes one-dimensional convolution between the input $\boldsymbol{v}$ and the kernels $\boldsymbol{f}$ as follows:

$$\boldsymbol{\rho} = \left\{ \boldsymbol{\rho}_\varepsilon = \boldsymbol{v} * \boldsymbol{f}_\varepsilon \;\middle|\; \varepsilon \in \left[1, n_\rho\right] \right\}, \tag{9}$$

where $\boldsymbol{\rho}_\varepsilon$ is the output of kernel $\boldsymbol{f}_\varepsilon$, and n_ρ represent the number of kernels. At last, we adopt an sigmoid layer as an output activation layer, given by $\boldsymbol{\sigma} = \left(1 + e^{-\boldsymbol{v}}\right)^{-1}$. We utilize the simplified form of JS divergence as the value function, which will be discussed later in this section.

In adversarial learning, the output samples $\boldsymbol{s}_G = (\boldsymbol{x}_G, \boldsymbol{y'})$ are fed into D for training alternatively. In IDS, the output samples $\boldsymbol{s}_G$ are mixed with the original samples to cope with the class imbalance problem.

3.1.2. Imbalanced adversarial learning

The procedure of learning includes three steps: data filtering, adversarial learning, and sample generating. When filtering data, we first draw a subset of minority classes $\boldsymbol{s'}$ as Eq. (6), and calculate the generating quantity $\boldsymbol{k}$ as Eq. (7).

After that, G and D optimize each other adversarially by playing a minimax two-player game. G aims at increasing the similarity between the real distribution $p_{data}(\boldsymbol{x})$ and the generative distribution $p_g(\boldsymbol{z})$, while D tries to differentiate the two distributions. Such a process can be described by training alternatively on a value function $\hat{V}(D, G)$, formalized as:

$$\min_G \max_D \hat{V}(D, G). \tag{10}$$

Based on the original value function (4), we realize $\hat{V}(D, G)$ with the samples of minority classes, taking the class labels as conditions [22], formulated as:

$$\begin{aligned} \hat{V}(D, G) = {} & \mathbb{E}_{\boldsymbol{x'}, \boldsymbol{y'} \sim p_{data}(\boldsymbol{x'}, \boldsymbol{y'})}[\log D(\boldsymbol{x'}, \boldsymbol{y'})] \\ & + \mathbb{E}_{\boldsymbol{y'} \sim p_{\boldsymbol{y'}}, \boldsymbol{z} \sim p_z(\boldsymbol{z})}[\log\left(1 - D(G(\boldsymbol{z}, \boldsymbol{y'}), \boldsymbol{y'})\right)], \end{aligned} \tag{11}$$

which intrinsically represents the conditional JS divergence between $p_{data}(\boldsymbol{x})$ and $p_g(\boldsymbol{z})$.

In each iteration, we first optimize D with fixed G and real samples, and then we optimize G with fixed D. As D and G are optimized alternatively, we can rephrase $\hat{V}(D, G)$ in terms of $\hat{V}(D)$ and $\hat{V}(G)$, for separate training as follows.

When optimizing D, we first draw m samples $\left\{\left(\boldsymbol{x}_i', \boldsymbol{y}_i'\right)\right\}_{i=1}^m$ from $p_{data}(\boldsymbol{x'}, \boldsymbol{y'})$, and draw m noises $\left\{\boldsymbol{z}_i\right\}_{i=1}^m$ from $p_z(\boldsymbol{z})$. Then, we can **maximize** the value function $\hat{V}(D)$ as follows:

$$\begin{aligned} \max_{\theta_D} \hat{V}(D) = \max_{\theta_D} \frac{1}{m} \sum_{i=1}^{m} \Big(& \log D\left(\boldsymbol{x}_i', \boldsymbol{y}_i'\right) \\ & + \log\left(1 - D\left(G\left(\boldsymbol{z}_i, \boldsymbol{y}_i'\right), \boldsymbol{y}_i'\right)\right)\Big), \end{aligned} \tag{12}$$

where θ_D notates the parameters of D. And we update θ_D by **ascending** on stochastic gradient $\triangledown_{\theta_D} \hat{V}(D)$ to finish the above optimization.

The optimization of G is almost identical to D. We draw m noises $\left\{\boldsymbol{z}_i\right\}_{i=1}^m$ from $p_z(\boldsymbol{z})$ again, and **minimize** the value function $\hat{V}(G)$ as follows:

$$\min_{\theta_G} \hat{V}(G) = \min_{\theta_G} \frac{1}{m} \sum_{i=1}^{m} \left(\log\left(1 - D\left(G\left(\boldsymbol{z}_i, \boldsymbol{y}_i'\right), \boldsymbol{y}_i'\right)\right)\right), \tag{13}$$

where θ_G notates the parameters of G. And we update θ_G by **descending** on stochastic gradient $\triangledown_{\theta_G} \hat{V}(G)$ to finish the above optimization. Moreover, we adopt the Adam algorithm [50] in the experiment to update θ_D and θ_G until D converges to 0.5.

After the adversarial learning, G can be utilized to generate samples for IDS, with classes $\boldsymbol{c}$ and noises $\boldsymbol{z}$. For each class c_τ in $\boldsymbol{c}$, G generates specific number of samples according to k_{c_τ} in $\boldsymbol{k}$. Eventually, IGAN merges the synthesized samples of different classes as $\hat{\boldsymbol{s}}_G$:

$$\hat{\boldsymbol{s}}_G = \bigcup_{c_\tau \in \boldsymbol{c}} \left\{ (G(\boldsymbol{z}_i, c_\tau), c_\tau) \;\middle|\; \boldsymbol{z}_i \sim p_z(\boldsymbol{z}) \right\} i = 1^{k_{c_\tau}}. \tag{14}$$

We summarize the overall filtering, learning, and generating procedures of IGAN in Algorithm 1 .

3.2. IGAN-based intrusion detection system (IGAN-IDS)

IGAN copes with the class imbalance problem in intrusion detection and simulates the unknown anomalies. An IGAN-based Intrusion Detection System (IGAN-IDS) is further built to perform integral detection.

3.2.1. System architecture

IGAN-IDS consists of three modules: feature extraction (FE), IGAN, and deep neural network (DNN). In Fig. 2, we show the system architecture of IGAN-IDS. First, the FE module transforms raw network attributes into latent feature vectors. Then, the IGAN module generates samples, as the previous Section 3.1. Finally, the DNN module is trained with balanced samples and executes intrusion detection on brand-new data. Generally speaking, IGAN-IDS takes network attributes $\boldsymbol{\alpha}$ as input and predicts their probability distributions $\boldsymbol{p}(\boldsymbol{y}|\boldsymbol{\alpha})$ over different intrusion classes.

Algorithm 1: IGAN, our proposed model.

Input: $\boldsymbol{s} = (\boldsymbol{x}, \boldsymbol{y})$, the original samples where $\boldsymbol{x}$ denotes feature vectors and $\boldsymbol{y}$ denotes class labels. r, the generated ratio. α_D, the learning rate of D. α_G, the learning rate of G.

Output: $\boldsymbol{s}_G$, the synthesized samples.

Parameter: t, iteration times of D per global iteration.

/* Imbalanced data filter */

$\boldsymbol{s}' \leftarrow \left\{ s' = (x', y') \mid s' \in \boldsymbol{s}, y' \neq \underset{c_\tau \in \boldsymbol{c}}{argmax}(n_{c_\tau}) \right\}$.

$\boldsymbol{k} \leftarrow \left\{ k_{c_\tau} = n_{c_\tau} \cdot r \mid c_\tau \in \boldsymbol{c} \right\}$.

/* Adversarial learning */

while D *has not converged to 0.5* **do**

  /* Optimization of the discriminator */

  for t *steps* **do**

    Sample $\left\{ (\boldsymbol{x}'_i, \boldsymbol{y}'_i) \right\}_{i=1}^{m}$ as a batch from $p_{data}(\boldsymbol{x}', \boldsymbol{y}')$.

    Sample $\left\{ \boldsymbol{z}_i \right\}_{i=1}^{m}$ as a batch from $p_z(\boldsymbol{z})$.

    $\eta_{\theta_D} \leftarrow \nabla_{\theta_D} \Big[\frac{1}{m} \sum_{i=1}^{m} \Big(\log D(\boldsymbol{x}'_i, \boldsymbol{y}'_i) + \log\big(1 - D(G(\boldsymbol{z}_i, \boldsymbol{y}'_i), \boldsymbol{y}'_i)\big) \Big) \Big]$.

    $\theta_D \leftarrow \theta_D + \alpha_D \cdot Adam(\theta_D, \eta_{\theta_D})$.

  end

  /* Optimization of the generator */

  Sample $\left\{ \boldsymbol{z}_i \right\}_{i=1}^{m}$ as a batch from $p_z(\boldsymbol{z})$.

  $\eta_{\theta_G} \leftarrow \nabla_{\theta_G} \big[\frac{1}{m} \sum_{i=1}^{m} \big(\log\big(1 - D(G(\boldsymbol{z}_i, \boldsymbol{y}'_i), \boldsymbol{y}'_i)\big) \big) \big]$.

  $\theta_G \leftarrow \theta_G - \alpha_G \cdot Adam(\theta_G, \eta_{\theta_G})$.

end

/* Generating samples */

$\hat{\boldsymbol{s}}_G \leftarrow \bigcup_{c_\tau \in \boldsymbol{c}} \left\{ (G(\boldsymbol{z}_i, c_\tau), c_\tau) \mid \boldsymbol{z}_i \in p_z(\boldsymbol{z}) \right\}_{i=1}^{k_{c_\tau}}$.

return $\hat{\boldsymbol{s}}_G$

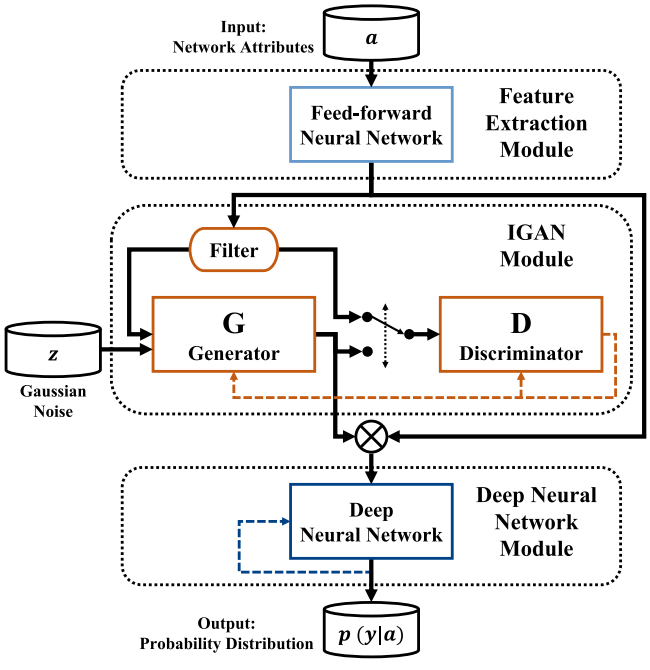

Fig. 2. System architecture of IGAN-IDS.

Table 1
Structure of the DNN module.

#	Layer	Size	Kernel	Activation
1	Fully-connected	256	–	Sigmoid
2	Convolutional	64	1×9	ReLU
3	Convolutional	64	1×9	ReLU
4	Dropout (rate = 0.2)	–	–	–
5	Fully-connected	32	–	Leaky ReLU
6	Fully-connected	$n_{\boldsymbol{c}}$	–	Softmax

3.2.2. Feature extraction module

The raw network attributes contain heterogeneous data from multiple sources, embodied in the mix of continuous and discrete data. Furthermore, there is a good deal of redundancy in the network attributes. Hence, the feature extraction module (FE) takes advantage of a Feed-forward Neural Network to transform them into latent feature vectors, as the top of Fig. 2. This module increases the availability of IGAN-IDS over various ad-hoc networks.

We implement the FE module with an embedding layer and a Multilayer Perceptron. First, the FE module embeds the discrete data in one-hot vectors, and discretizes continuous data and encodes them in a one-hot manner as well. Later, all the one-hot data are concatenated and fed into a Multilayer Perceptron to extract latent features. The FE module can be formulated as $\boldsymbol{x} = FE(\boldsymbol{\alpha})$, where $\boldsymbol{\alpha}$ and $\boldsymbol{x}$ denote the raw network attributes and latent feature vectors respectively.

3.2.3. IGAN module

In the previous Section 3.1, we have described the IGAN module in detail. We put the IGAN module into practice to cope with the class imbalance problem in intrusion detection. As shown in the middle of Fig. 2, the IGAN module takes the latent feature vectors from the FE module as input, together with Gaussian noise, and generate samples for minority classes. The IGAN module can be formulated as $\hat{\boldsymbol{s}}_G = IGAN(\boldsymbol{x}, \boldsymbol{y})$, where $\boldsymbol{x}$ is the latent feature vectors from the FE module and $\boldsymbol{y}$ is the corresponding class labels.

3.2.4. Deep neural network module

The deep neural network module (DNN) is created to execute intrusion detection, as Fig. 2. In the training phase, the DNN module takes the samples mixed with real ones and synthesized ones as input. In the predicting phase, the DNN module takes brand-new samples as input and output their probability distributions $\boldsymbol{p}(\boldsymbol{y}|\boldsymbol{\alpha})$ over different intrusion classes. Table 1 shows the detailed structure of the DNN module.

We utilize a Convolutional Neural Network (CNN) to furnish reliable classification ability. The model structure is designed referring to [16], and later we fine-tune the parameters with grid search on the validation set. The fully-connected layers learn global information, and the convolutional layers capture local features. In more detail, we use a fully-connected layer with 256 neurons as the input layer, whose activation function is sigmoid. After that, we employ two convolutional layers with 64 kernels, whose activation function of these layers is ReLU. The output of convolutional layers applies a random dropout, with a rate of 0.2, to avoid overfitting. Moreover, we deploy a 32-neurons fully-connected layer, with Leaky ReLU as activation. The size of the output layer equals the number of classes $n_{\boldsymbol{c}}$. Finally, we practice a softmax function to turn the output vectors into probability distributions:

$$\boldsymbol{p}(\boldsymbol{y} \mid \boldsymbol{\alpha}) = \boldsymbol{p}(y = c_\tau \mid \boldsymbol{v}) = \frac{e^{v_\tau}}{\sum_{v_\tau \in \boldsymbol{v}} e^{v_\tau}}, \tag{15}$$

where v_τ is the τth element of the input vector $\boldsymbol{v}$ from the upper layer. $\boldsymbol{p}(y = c_\tau \mid \boldsymbol{v})$ is the final output of the whole IGAN-IDS, representing the probability that the given sample belongs to class c_τ.

Table 2
Description of the NSL-KDD attributes.

Attributes	Description
1–9	Basic features of network connections.
10–22	Content-related traffic features.
23–31	Time-related traffic features.
32–41	Host-based traffic features.

Table 3
Description of the UNSW-NB15 attributes.

Attributes	Description
1–13	Basic features of network connections.
14–21	Content-related traffic features.
22–30	Time-related traffic features.
31–35	General purpose traffic features.
36–42	Connection-based traffic features.

Table 4
Description of the CICIDS2017 attributes.

Attributes	Description
1–6	Basic features of network connections.
7–14	Features of network packets.
15–20	Features of network flows.
21–43	Statistic of network flows.
44–62	Content-related traffic features.
63–66	Features of network subflows.
67–78	General purpose traffic features.

4. Experiment and discussion

In the following, we conduct experiments to evaluate the performance of IGAN-IDS. We compare IGAN-IDS against conventional methods, class balancing methods, and state-of-the-art methods, showing that IGAN-IDS outperforms the state-of-the-art approaches. We further discuss the robustness of IGAN-IDS with different generated ratios and different imbalance ratios. Finally, we assess the effectiveness of each module in IGAN-IDS through an ablation study.

4.1. Benchmark dataset

The NSL-KDD dataset [51], the UNSW-NB15 dataset [52] and the CICIDS2017 dataset [20] are adopted to evaluate IGAN-IDS. They are widely-used datasets to evaluate IDS, where NSL-KDD holds 148,517 samples, UNSW-NB15 includes 2,540,044 ones, and CICIDS2017 contains 2,827,829 ones.

NSL-KDD is one of the most classical benchmark datasets for evaluating intrusion detection, which helps the researchers compare the efficiency of various approaches. Each sample in NSL-KDD includes 41 attributes listed in Table 2. These attributes, divided into four categories, include basic features, content-related features, time-related features, and host-based features. And NSL-KDD was partitioned into a training set and a test set in advance, namely KDDTrain^+ and KDDTest^+, respectively.

Moustafa et al. [52] brought forward the UNSW-NB15 dataset in 2015 as a hybrid of various anomalies. Different intrusion detection mechanisms can be well tested by the multiple types of attacks in UNSW-NB15. Each sample in UNSW-NB15 contains 42 attributes, regarding basic features, content-related features, time-related features, general purpose features, and connection-based features, as listed in Table 3.

CICIDS2017 is a latest updated IDS dataset that includes five common network attack flows. Besides, up to 2,827,829 samples in CICIDS2017 can evaluate the performance of IDS in large-scale scenarios. It contains 78 attributes divided into seven categories, as listed in Table 4.

Table 5
Distribution of the datasets.

Dataset	Class	Samples	Imbalance ratio
NSL-KDD	Normal	77,054	–
	DoS	53,385	1.443
	Probe	14,077	5.474
	R2L	3749	20.553
	U2R	252	305.770
UNSW-NB15	Normal	2,218,761	–
	Generic	215,481	10.297
	Exploits	44,525	49.832
	Fuzzers	24,246	91.510
	DoS	16,353	135.679
	Reconnaissance	13,987	158.630
	Analysis	2677	828.824
	Backdoor	2329	952.667
	Shellcode	1511	1,468.406
	Worms	174	12,751.500
CICIDS2017	Benign	2,271,320	–
	DoS	379,737	5.587
	Port Scan	158,804	13.357
	Brute Force	13,832	153.345
	Web Attack	2180	972.966
	Bot	1956	1,084.389

However, NSL-KDD, UNSW-NB15, and CICIDS2017 are all typical imbalanced datasets. The imbalance ratios of NSL-KDD range from 1.44 to 305.77, the imbalance ratios of UNSW-NB15 range from 10.30 to 12,751.50, and the imbalance ratios of CICIDS2017 range from 5.59 to 1,084.39, as shown in Table 5. Furthermore, in NSL-KDD, 23 species of anomalies emerged in the training set, while 38 species appeared in the testing set. The class imbalance data and the emergence of 15 unknown anomalies embody the challenge of intrusion detection in reality.

We perform 5-class classification on NSL-KDD, 10-class classification on UNSW-NB15, and 6-class classification on CICIDS2017, rather than binary classification. We formulate intrusion detection as a multiclass classification problem and conduct multiple One-Vs-All classifications. In each classification, we see one specified class as the positive class, while the others as the negative class.

4.2. Evaluation metrics

To quantitatively evaluate IGAN-IDS, we adopt Accuracy, Precision, Recall, and F1 Score as the primary metrics. Besides, the Receiver Operating Characteristic (ROC) curve and the area under the ROC curve (AUC) are applied to reflect the generalization ability.

Accuracy acts as a metric to indicate the overall performance of IGAN-IDS. And for each class, Precision quantifies the specific classifying ability, while Recall reveals the specific detection rate. The definition of these metrics can be given by

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}, \tag{16}$$

$$Precision = \frac{TP}{TP + FP}, \tag{17}$$

$$Recall = \frac{TP}{TP + FN}, \tag{18}$$

where TP, TN, FP, and FN indicate True Positive, True Negative, False Positive and False Negative, respectively.

F1 Score is the harmonic mean of Precision and Recall, considering both the classifying ability and the detection rate. As a statistical formulation, the F1 Score is given by (19).

$$F1 = \frac{2}{Precision^{-1} + Recall^{-1}} = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}. \tag{19}$$

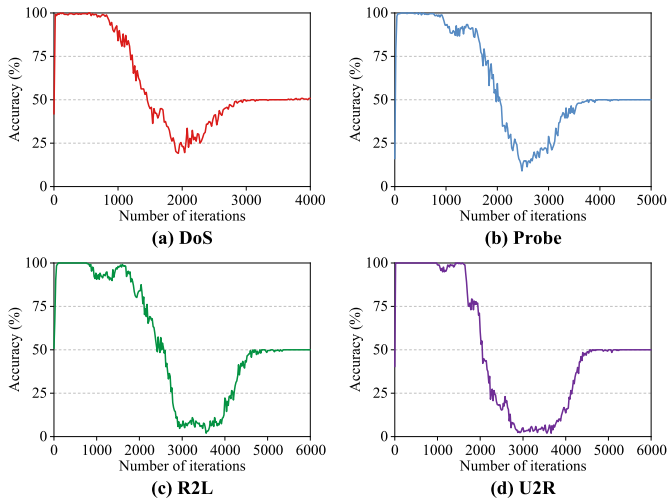

Fig. 3. Convergence curve of discriminator D.

The ROC curve plots the True Positive Rate (TPR) against the False Positive Rate (FPR), for all possible thresholds on the result from the model. The AUC is further to analyze the ROC curve by calculating the area under the curve. The AUC of a perfect model will reach 1, while the AUC of random classification results in 0.5. Therefore, the higher the AUC, the better the model performs.

4.3. Experiment procedure

As mentioned in Section 3.2, the raw network attributes are first fed into the FE module. The FE module contains an embedding layer and an MLP, where the embedding layer integrates the attributes and the MLP extract the features. The dimension of the embedding layer is set to 356, which depends on the reality of common network attributes. As for the MLP, we implement it with two fully-connected layers whose dimensions both are 128, with a sigmoid activation function as the output layer. The hyper-parameters of MLP are tuned with grid search on the validation set, by which the MLP can efficiently extract features and avoid unnecessary overhead at the same time. The feature vectors extracted from the FE module are fed into the IGAN module.

With regards to the IGAN module, the architecture of discriminator D and generator G is relatively flexible and can be set according to specific situations. We make a comprehensive reference to [16,21,22] to determine the structure of IGAN, and the parameters of IGAN are tuned with grid search on the validation set. With the selected parameters, IGAN can realize adequate expression ability with the lowest overhead. We build a D in the structure of MLP, composed of three fully-connected layers whose dimensions are set to 256, 128 and 64. The dimensions of fully-connected layers in G are all 256, while 64 kernels constitute the convolutional layer, and the kernel size is 16. The learning rates of IGAN (denoted as α_D and α_G in Algorithm 1) are both set to 0.00005. The batch size (denoted as m) is 128 and in each iteration we separately optimize D and G once ($t = 1$). The generated ratio (denoted as r) is set to 1:1. The IGAN module is trained until D converges to 0.5 on all classes, meaning the optimization of IGAN has completed.

Coping with the class imbalance in NSL-KDD, the IGAN module generates samples for four minority classes, including DoS, Probe, R2L, and U2R. Based on the Accuracy of each class, the convergence curves of D is shown in Fig. 3. From the curves, we observe that the IGAN module has been well optimized to generate samples for DoS, Probe, R2L, and U2R, after 4000, 5000, 6000 and 6000 iterations, respectively. The optimizing procedures on UNSW-NB15 and CICIDS2017 are the same and result in similar convergence curves.

Taking the curve of DoS (Fig. 3(a)) as an example, in the first 1000 iterations, G was not capable of generating representative samples, wherefore D could easily distinguish the synthesized samples from the real ones. Thus, D could reach almost 100% on Accuracy at the beginning. In the 1000th to 2000th iterations, the Accuracy of D approximately dropped from 100% to 20%, because the generative ability of G kept increasing and made it hard for D to differentiate synthesized and real samples. In the last 2000 iterations, D and G kept optimizing each other until they reached an equilibrium, where D converged to 0.5 and lost discrimination ability.

Fig. 3 (b) indicates the curve of Probe, where the Accuracy reached almost 100% in the first 1000 iterations since G was weak at generating samples. Later in the 1000th to 2500th iterations, the Accuracy approximately dropped from 100% to 10% because the generative ability of G kept increasing. In the last 2500 iterations, D and G kept optimizing each other until D converged to 0.5. As for R2L and U2R (Fig. 3(c) and (d)), the Accuracy of D attained almost 100% in the first 2000 iterations for G was inadequate for generating samples. In the 2000th to 4000th iterations, the Accuracy approximately dropped from 100% to 5% as the generative ability of G kept increasing. In the last 2000 iterations, D and G kept optimizing each other, and D converged to 0.5 finally.

The DNN module takes the synthesized samples and real samples as input for training, as described in Section 3.2. After well optimized, the DNN module executes the final intrusion detection on brand-new data. We take the Categorical Cross-Entropy as the loss function and apply the Adam algorithm to optimize DNN with a learning rate of 0.00005.

4.4. Comparative study

Based on the above datasets and metrics, we employ three types of methods for comparative study, including conventional methods, class balancing methods, and state-of-the-art methods. The comparative results (Table 6) illustrate that IGAN-IDS outperforms the state-of-the-art approaches.

4.4.1. Baseline methods

First, we consider several conventional methods as follows for comparison. Naive Bayes and Decision Tree [8] are both well-known learning methods, which can provide satisfactory performance with low overhead. Random Forest [9] is an ensemble learning method constructed by a multitude of Decision Trees but has stronger generalization ability than Decision Tree. Support Vector Machine (SVM) [10] is a classic and efficient classification method, but not applicable with big data [53]. Multilayer Perceptron (MLP) [15] is a fundamental deep learning model that has a stable classification ability.

Concerning the class balancing methods, we make use of random under-sampling (RUS), random over-sampling (ROS) [30], and SMOTE [36] algorithm, integrating with SVM and MLP. Concretely, we utilize the class balancing methods to generate samples until the sample quantities of various classes are identical. And then we feed the balanced samples into SVM or MLP, to execute intrusion detection. We mark these methods as RUS + SVM, RUS + MLP, ROS + SVM, ROS + MLP, SMOTE + SVM and SMOTE + MLP.

To demonstrate the superiority of IGAN-IDS, we compare it with several state-of-the-art methods.

- Convolution Neural Network (CNN) is one of the state-of-the-art representation learning methods [16]. In more detail, it has two convolutional layers that contain 32 and 64 kernels, respectively.
- Fuzziness-based Neural Network (NN) [25] is a semi-supervised learning approach, which can improve the generalization of

Table 6
Comparison results between IGAN-IDS and different methods (%).

Model	NSL-KDD			UNSW-NB15			CICIDS2017		
	Accuracy	F1 Score	AUC	Accuracy	F1 Score	AUC	Accuracy	F1 Score	AUC
Naive Bayes	73.55	72.31	91.76	61.80	65.27	90.13	93.90	93.53	97.49
Decision Tree	77.89	75.25	81.71	73.52	76.36	86.56	99.62	99.57	99.56
Random Forest	77.20	73.23	86.68	74.35	77.28	95.18	99.79	99.78	99.98
SVM	72.85	68.84	84.46	68.49	70.13	95.15	96.97	96.99	98.98
MLP	78.97	75.40	90.24	78.32	76.98	96.70	99.48	99.39	99.95
RUS + SVM	73.57	70.11	85.35	67.16	70.45	94.98	96.45	96.55	98.96
RUS + MLP	76.66	72.38	92.34	77.27	76.21	96.67	99.46	99.42	99.79
ROS + SVM	73.34	69.90	84.96	68.32	70.00	95.08	96.98	97.04	98.96
ROS + MLP	78.10	74.18	91.44	76.13	76.97	96.29	99.55	99.55	99.88
SMOTE + SVM	79.23	78.36	90.86	71.50	73.77	94.44	97.00	97.04	98.97
SMOTE + MLP	77.47	75.18	93.01	79.59	80.10	96.89	99.33	99.34	99.91
CNN	78.33	74.75	93.72	80.52	76.61	96.72	99.48	99.44	99.94
Fuzziness-based NN	75.33	70.58	93.32	81.21	78.58	97.02	99.61	99.57	99.83
LSSVM + MIFS ($\beta = 0.3$)	78.20	72.76	84.39	76.83	77.43	96.87	98.76	98.67	99.73
LSSVM + FMIFS	75.67	73.67	85.84	77.18	77.65	96.81	99.51	99.48	99.83
IGAN-IDS	**84.45**	**84.17**	**95.55**	**82.53**	**82.86**	**97.09**	**99.79**	**99.79**	**99.98**

Note: F1 Score and AUC are the average of multiple classes, weighted by the number of samples in each class.

IDS through a fuzziness categorization. It needs training twice, where the first training is to categorize the fuzziness of samples while the second is training with reconstructed data.

- Amiri et al. [13] introduced a Mutual Information based Feature Selection (MIFS) before the Least Square SVM (LSSVM). It performs greedy selection on features, with a determined parameter β telling the expected number of selected features. According to the empirical evidence in intrusion detection, 0.3 is the optimal value of β.
- Based on LSSVM + MIFS, Ambusaidi et al. [14] further proposed a Flexible MIFS (FMIFS) method, serving as an adaptive features selection approach without any empirically determined parameter.

We compare our proposed IGAN-IDS against the above 15 methods to evaluate the performance. Unless otherwise stated, all hyper-parameters of the above methods are tuned by grid searching on the validation dataset.

4.4.2. Comparison results

The comparison results between IGAN-IDS and different methods are summarized in Table 6. IGAN-IDS outperforms other methods on NSL-KDD, UNSW-NB15, and CICIDS2017.

Among the conventional methods, MLP attains a considerable performance, especially on Accuracy (78.97% on NSL-KDD and 78.32% on UNSW-NB15) and AUC (90.24% on NSL-KDD and 96.70% on UNSW-NB15). As a deep learning model, MLP has a proven classification ability, but still has difficulties in dealing with class-imbalanced data. Naive Bayes gets higher AUC on NSL-KDD (91.76%) since it is a generative model, less affected by class imbalance. But, its inadequate expression ability limits its overall performance. Random Forest also gets good results on UNSW-NB15 (74.35% on Accuracy and 95.18% on AUC) and CICIDS2017 (99.79% on Accuracy and 99.98% on AUC), showing its generalization ability as an ensemble model. However, the above results reflect the flaws of conventional methods: insufficient expression ability on data and weak capacity to cope with class-imbalanced data.

In line with our expectations, SVM (SMOTE) performs better than SVM on both NSL-KDD and UNSW-NB15, because the class balancing method SMOTE helps counter the class imbalance problem. Besides, the method can learn more about the minority classes from the samples generated by SMOTE, enhancing its generalization ability. However, when comparing MLP (SMOTE) with MLP on NSL-KDD, the metrics other than AUC slightly decreases. It reflects the trade-off between the classification ability and generalization ability. As the classification ability is of great significance for intrusion detection, SMOTE is not capable enough.

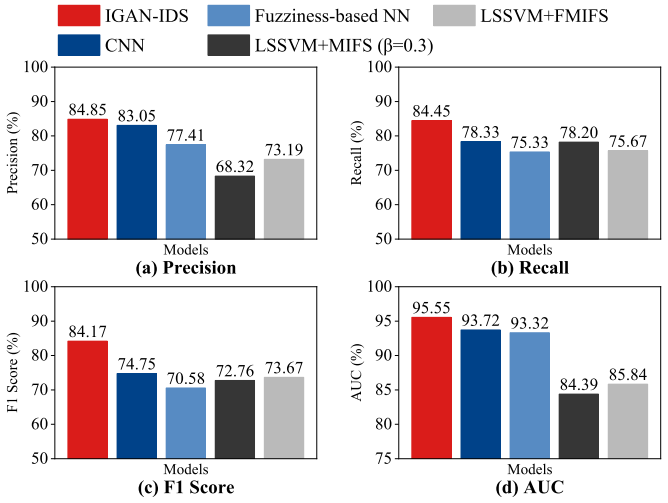

Fig. 4. Comparison results between IGAN-IDS and state-of-the-art methods on NSL-KDD.

We compare the performance of IGAN-IDS against state-of-the-art methods in detail on NSL-KDD, as shown in Fig. 4. IGAN-IDS achieves at least 1%, 6%, 10%, 1% improvement on Precision, Recall, F1 Score, and AUC. The better performance of IGAN-IDS is due to the high representative synthesized samples for the minority classes from IGAN, alleviating the class imbalance problem and simulating the unknown anomalies.

We further plot the ROC curves of IGAN-IDS and state-of-the-art methods on NSL-KDD, as Fig. 5. The ROC curve of IGAN-IDS is the closest one to the upper left corner, indicating a better generalization ability against the other methods.

All the results reported above demonstrate that IGAN-IDS outperforms its competitors. We can conclude that IGAN effectively handles the class imbalance problem by generating samples for minority classes, and IGAN-IDS is capable of efficient intrusion detection.

In the following, we further examine the robustness of IGAN-IDS with different generated ratios and different imbalance ratios, based on NSL-KDD.

Table 7
Accuracy (%) of IGAN-IDS with different generated ratios.

Ratio (r)	0	0.5: 1	1: 1	1.5: 1	2: 1
Accuracy	79.80	84.07	84.45	82.78	82.03

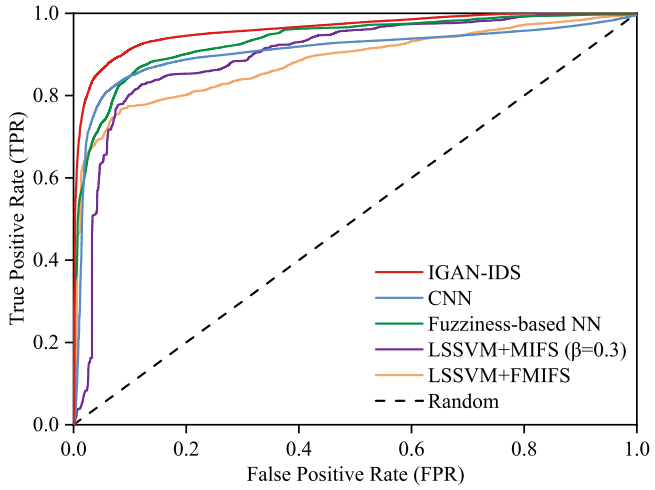

Fig. 5. ROC curves of IGAN-IDS and state-of-the-art methods on NSL-KDD.

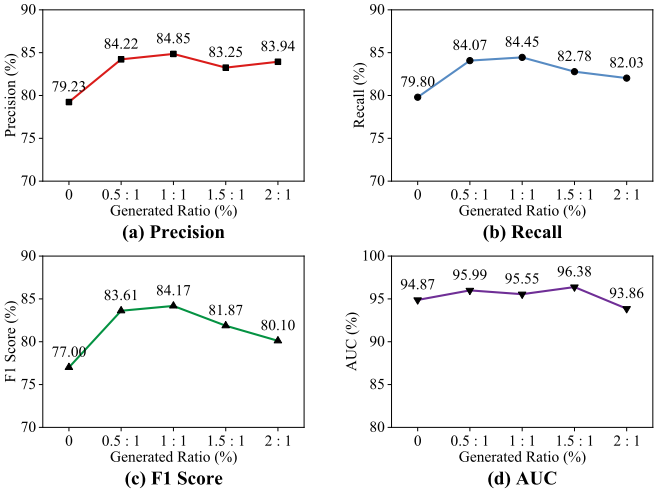

Fig. 6. Performance of IGAN-IDS with different generated ratios.

4.4.3. IGAN-IDS with different generated ratios

To verify the effect of synthesized samples, we assess IGAN-IDS with a series of experiments using different generated ratios (denoted as r in Algorithm 1).

The Accuracy of IGAN-IDS with different generated ratios is reported in Table 7. Compared with the result without synthesized samples ($r = 0$), we can observe that IGAN-IDS performs much better (4% higher on Accuracy) when generating samples during training, illustrating the positive effect of IGAN.

We further compare the detail performance of IGAN-IDS with different generated ratios, as Fig. 6. From Table 7 and Fig. 6, we can observe that IGAN-IDS gets highest Accuracy, Precision, Recall and F1 Score with generated ratio $r = 1 : 1$, while the highest AUC is achieved with $r = 1.5 : 1$. In our study, it's suggested that the generated ratio should be controlled within the range from $r = 0.5 : 1$ to $r = 1.5 : 1$ for stable performance.

To conclude, the IGAN indeed brings performance increase to intrusion detection through generating samples, and the generated ratio should be carefully controlled.

Table 8
Performance (%) of IGAN-IDS with different imbalance ratios, under-sampling the minority classes with certain sampling rates in KDDTrain$^+$ for training, while testing on complete KDDTest$^+$.

Rate (%)	100	90	80	70	60	50
Accuracy	84.45	82.67	82.55	82.02	82.02	82.01
Precision	84.85	83.73	83.60	83.95	83.95	83.94
Recall	84.45	82.67	82.55	82.02	82.02	82.01
F1 Score	84.17	81.52	81.43	80.09	80.08	80.08
AUC	95.55	96.25	96.24	93.83	92.83	92.83

Table 9
Results of the ablation study (%).

Model	Module			Accuracy
	FE	DNN	IGAN	
IGAN-IDS	√	√	√	**84.45**
(1) w/o IGAN	√	√	–	79.80
(2) w/o FE	–	√	√	82.01
(3) DNN Only	–	√	–	79.22
(4) FE Only	√	–	–	78.86

4.4.4. IGAN-IDS with different imbalance ratios

We evaluate IGAN-IDS with different imbalance ratios between the majority class and the minority classes. The imbalance ratios are manually set by randomly under-sampling the minority classes with different sampling rates. For example, the original sample quantity of class Probe in NSL-KDD is 14,077. When we under-sample it with a sampling rate of 50%, 7038 samples are randomly selected, and the imbalance ratio is doubled from 5.4738 to 10.9476. We only change the imbalance ratios of the training set for training, while the test set remains complete for testing.

The Performance of IGAN-IDS with different imbalance ratios is reported in Table 8. From the results, IGAN-IDS performs relatively stable with different imbalance ratios, since IGAN can generate samples of minority classes. However, with the imbalance ratio increasing, the performance of IGAN-IDS slightly drops. When the imbalance ratio becomes higher, less information is provided to IGAN, disabling IGAN to generate representative samples for the whole data distribution.

As our expectations, when increasing the imbalance ratios, all metrics keep the trends of dropping. However, even when the imbalance ratio is doubled (sampling rate 50%), IGAN-IDS still shows a considerable performance, verifying the robustness of IGAN-IDS.

Overall, IGAN-IDS can remain aggressive performance under severe class imbalance.

4.5. Ablation study

For a thorough analysis, we conduct an ablation study on IGAN-IDS to analyze the effectiveness of each module. The detail of the ablation study based on NSL-KDD is listed as follows.

(1) w/o IGAN: We remove the IGAN module from IGAN-IDS but keep the FE module and the DNN module. Since there is no sample generated, the results should be identical to IGAN-IDS with the generated ratio of $r = 0$.
(2) w/o FE: The FE module is removed from IGAN-IDS, where the raw network attributes are fed into IGAN and DNN after being embedded into one-hot vectors.
(3) DNN Only: We only keep the DNN module to execute intrusion detection, and the raw network attributes are embedded into one-hot vectors in advance.
(4) FE Only: We see the FE module as a classifier by adding a softmax function on the last layer.

Table 10
Detail performance (%) of IGAN-IDS in ablation study.

Model	Precision	Recall	F1 Score	AUC
IGAN-IDS	**84.85**	**84.45**	**84.17**	**95.55**
(1) w/o IGAN	79.23	79.80	77.00	94.87
(2) w/o FE	83.94	82.01	80.08	93.82
(3) DNN Only	82.71	79.22	77.27	92.31
(4) FE Only	82.58	78.86	76.67	90.79

The results of the ablation study are displayed in Table 9. Comparing IGAN-IDS with Model (1), or comparing Model (2) with Model (3), we can conclude that the IGAN module can contribute to intrusion detection. It is because the IDS can learn the information about minority classes more thoroughly, via IGAN's balancing the samples.

The effectiveness of FE also can be demonstrated by comparing IGAN-IDS with Model (2) or comparing Model (1) with Model (3). If we use the FE module only, we can still reach an Accuracy close to state-of-the-art methods. When we remove the FE module, the Accuracy drops because the model fails to extract high-dimensional features.

We further analyze the detail performance of IGAN-IDS in ablation study as Table 10. The detailed result meets the analysis of the IGAN module and the FE module mentioned above. Moreover, Model (1) is the combination of Model (3) and Model (4), achieving higher on AUC but lower on Precision. It is because Model (1) is more intricate than Model (3) or (4). Model (3) or (4) can be well optimized with the existing data, while Model (1) needs more samples for training. Thus, Model (1) has better generalization ability (higher AUC) but appears to be weaker in classification. This phenomenon clears that the synthesized samples are necessary for better performance, especially the ones of minority classes.

5. Conclusion

Security is one of the most critical issues in cyber systems, especially in dynamic and decentralized ad-hoc networks. Therefore, intrusion detection plays a significant role in network security, where the class-imbalanced data is a challenging problem.

In this paper, we proposed a novel Imbalance Generative Adversarial Network (IGAN), to tackle the class imbalance problem in intrusion detection by generating samples. IGAN introduces a filter on data to ensure only generating samples for minority classes, which is an improvement over GAN. Further, we build an IGAN-based Intrusion Detection System (IGAN-IDS) to cope with class imbalance intrusion detection. Compared with the existing intrusion detection methods, we adopt IGAN to generate representative samples for improving the performance of IDS. Besides, we apply IGAN-IDS on three datasets, and the comparative study demonstrates that our approach outperforms the other 15 competitors, including the state-of-the-art ones. The additional studies on robustness and ablation show the powerful capacity of IGAN-IDS to counter class imbalance intrusion detection.

In our future work, we will consider the reduction of building time via online learning (e.g., Siamese Neural Network [54], deep reinforcement learning [55]). With a Siamese Neural Network, Daudt et al. [56] performed change detection, which largely reduced the time overhead and achieved near state-of-the-art performance. Therefore, we can consider applying Siamese Neural Network for more efficient intrusion detection. Further, Mnih et al. [55] realized human-level control based on deep reinforcement learning, showing adaptability towards the real-time tasks, which we may utilize in intrusion detection as well. Also, we will study the usage of attention mechanisms [57] to extract attentive information from various traffic patterns, enhancing the overall semantic comprehension of network attributes.

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.

Acknowledgements

This work was financially supported by Shenzhen Key Laboratory Project (ZDSYS201802051831427) and the project "PCL Future Regional Network Facilities for Large-scale Experiments and Applications"

Supplementary material

Supplementary material associated with this article can be found, in the online version, at10.1016/j.adhoc.2020.102177

References

[1] L. Zhou, Z.J. Haas, Securing ad hoc networks, IEEE Netw. 13 (6) (1999) 24–30, doi:10.1109/65.806983.
[2] K. Grahn, M. Westerlund, G. Pulkkis, Analytics for network security: a survey and taxonomy, in: Information Fusion for Cyber-Security Analytics, volume 691, Springer, 2017, pp. 175–193, doi:10.1007/978-3-319-44257-0_8.
[3] Q. Chen, R.A. Bridges, Automated behavioral analysis of malware: a case study of WannaCry ransomware, in: 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA), IEEE, Cancun, Mexico, 2017, pp. 454–460, doi:10.1109/ICMLA.2017.0-119.
[4] D. Kwon, H. Kim, J. Kim, S.C. Suh, I. Kim, K.J. Kim, A survey of deep learning-based network anomaly detection, Cluster Comput. 22 (1) (2017) 1–13, doi:10.1007/s10586-017-1117-8.
[5] R. Chalapathy, S. Chawla, Deep learning for anomaly detection: a survey, arXiv preprint arXiv:1901.03407(2019) 1–50.
[6] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Maciá-Fernández, E. Vázquez, Anomaly-based network intrusion detection: techniques, systems and challenges, Comput. Secur. 28 (1–2) (2009) 18–28, doi:10.1016/j.cose.2008.08.003.
[7] M. Panda, M.R. Patra, Network intrusion detection using Naive Bayes, Int. J. Comput. Sci. Netw. Secur. 7 (12) (2007) 258–263.
[8] N.B. Amor, S. Benferhat, Z. Elouedi, Naive Bayes vs decision trees in intrusion detection systems, in: Proceedings of the 2004 ACM Symposium on Applied Computing, in: SAC '04, ACM, Nicosia, Cyprus, 2004, pp. 420–424, doi:10.1145/967900.967989.
[9] J. Zhang, M. Zulkernine, A. Haque, Random-forests-based network intrusion detection systems, IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.) 38 (5) (2008) 649–659, doi:10.1109/TSMCC.2008.923876.
[10] Y. Wang, J. Wong, A. Miner, Anomaly intrusion detection using one class SVM, in: Proceedings from the Fifth Annual IEEE SMC Information Assurance Workshop, 2004., IEEE, West Point, NY, USA, 2004, pp. 358–364, doi:10.1109/IAW.2004.1437839.
[11] M.A.M. Hasan, M. Nasser, B. Pal, S. Ahmad, Support vector machine and random forest modeling for intrusion detection system (IDS), J. Intell. Learn. Syst. Appl. 6 (1) (2014) 45, doi:10.4236/jilsa.2014.61005.
[12] H. Deng, Q.-A. Zeng, D.P. Agrawal, SVM-based intrusion detection system for wireless ad hoc networks, in: 2003 IEEE 58th Vehicular Technology Conference. VTC 2003-Fall (IEEE Cat. No. 03CH37484), volume 3, IEEE, Orlando, FL, USA, 2003, pp. 2147–2151, doi:10.1109/VETECF.2003.1285404.
[13] F. Amiri, M.R. Yousefi, C. Lucas, A. Shakery, N. Yazdani, Mutual information-based feature selection for intrusion detection systems, J. Netw. Comput. Appl. 34 (4) (2011) 1184–1199, doi:10.1016/j.jnca.2011.01.002.
[14] M.A. Ambusaidi, X. He, P. Nanda, Z. Tan, Building an intrusion detection system using a filter-based feature selection algorithm, IEEE Trans. Comput. 65 (10) (2016) 2986–2998, doi:10.1109/TC.2016.2519914.
[15] M. Moradi, M. Zulkernine, A neural network based system for intrusion detection and classification of attacks, in: Proceedings of the IEEE International Conference on Advances in Intelligent Systems - Theory and Applications, IEEE, Lux-embourg-Kirchberg, Luxembourg, 2004, pp. 15–18.
[16] Z. Li, Z. Qin, K. Huang, X. Yang, S. Ye, Intrusion detection using convolutional neural networks for representation learning, in: International Conference on Neural Information Processing, Springer, Guangzhou, China, 2017, pp. 858–866, doi:10.1007/978-3-319-70139-4_87.
[17] A. Chawla, B. Lee, S. Fallon, P. Jacob, Host based intrusion detection system with combined CNN/RNN model, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, Dublin, Ireland, 2018, pp. 149–158, doi:10.1007/978-3-030-13453-2_12.

[18] S. Aljawarneh, M. Aldwairi, M.B. Yassein, Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model, J. Comput. Sci. 25 (2018) 152–160, doi:10.1016/j.jocs.2017.03.006.
[19] Y. Yang, K. Zheng, C. Wu, Y. Yang, Improving the classification effectiveness of intrusion detection by using improved conditional variational autoencoder and deep neural network, Sensors 19 (11) (2019) 2528, doi:10.3390/s19112528.
[20] I. Sharafaldin, A.H. Lashkari, A.A. Ghorbani, Toward generating a new intrusion detection dataset and intrusion traffic characterization, in: ICISSP, INSTICC, Funchal, Madeira, Portugal, 2018, pp. 108–116, doi:10.5220/0006639801080116.
[21] I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Proceedings of the 27th International Conference on Neural Information Processing Systems, in: NIPS'14, volume 2, MIT Press, Montreal, Canada, 2014, pp. 2672–2680.
[22] J. Gauthier, Conditional generative adversarial nets for convolutional face generation, Class Project Stanford CS231N 2014 (5) (2014) 2.
[23] S. Revathi, A. Malathi, A detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection, Int. J. Eng. Res. Technol. (IJERT) 2 (12) (2013) 1848–1853.
[24] L. Dhanabal, S. Shantharajah, A study on NSL-KDD dataset for intrusion detection system based on classification algorithms, Int. J. Adv. Res. Comput. Commun. Eng. 4 (6) (2015) 446–452, doi:10.17148/IJARCCE.2015.4696.
[25] R.A.R. Ashfaq, X.-Z. Wang, J.Z. Huang, H. Abbas, Y.-L. He, Fuzziness based semi-supervised learning approach for intrusion detection system, Inf. Sci. 378 (1) (2017) 484–497, doi:10.1016/j.ins.2016.04.019.
[26] A.-U.-H. Qureshi, H. Larijani, N. Mtetwa, A. Javed, J. Ahmad, et al., RNN-ABC: a new swarm optimization based technique for anomaly detection, Computers 8 (3) (2019) 59, doi:10.3390/computers8030059.
[27] I. Ahmad, M. Basheri, M.J. Iqbal, A. Rahim, Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection, IEEE Access 6 (2018) 33789–33795, doi:10.1109/ACCESS.2018.2841987.
[28] N. Japkowicz, S. Stephen, The class imbalance problem: a systematic study, Intell. Data Anal. 6 (5) (2002) 429–449, doi:10.3233/IDA-2002-6504.
[29] X. Guo, Y. Yin, C. Dong, G. Yang, G. Zhou, On the class imbalance problem, in: 2008 Fourth International Conference on Natural Computation, volume 4, IEEE, Jinan, China, 2008, pp. 192–201, doi:10.1109/ICNC.2008.871.
[30] A. Puri, M.K. Gupta, Comparative analysis of resampling techniques under noisy imbalanced datasets, in: 2019 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT), volume 1, IEEE, GHAZIABAD, India, India, 2019, pp. 1–5, doi:10.1109/ICICT46931.2019.8977650.
[31] S. Rodda, U.S.R. Erothi, Class imbalance problem in the network intrusion detection systems, in: 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), IEEE, Chennai, India, 2016, pp. 2685–2688, doi:10.1109/ICEEOT.2016.7755181.
[32] V. Engen, J. Vincent, K. Phalp, Enhancing network based intrusion detection for imbalanced data, Int. J. Knowl. Based Intell. Eng. Syst. 12 (5–6) (2008) 357–367, doi:10.3233/KES-2008-125-605.
[33] L. Kuang, M. Zulkernine, An anomaly intrusion detection method using the CSI-KNN algorithm, in: Proceedings of the 2008 ACM Symposium on Applied Computing, ACM, Fortaleza, Ceara, Brazil, 2008, pp. 921–926, doi:10.1145/1363686.1363897.
[34] R. Abdulhammed, M. Faezipour, A. Abuzneid, A. AbuMallouh, Deep and machine learning approaches for anomaly-based intrusion detection of imbalanced network traffic, IEEE Sens. Lett. 3 (1) (2018) 1–4, doi:10.1109/LSENS.2018.2879990.
[35] D.A. Cieslak, N.V. Chawla, A. Striegel, Combating imbalance in network intrusion datasets., in: 2006 IEEE International Conference on Granular Computing, IEEE, Atlanta, GA, USA, 2006, pp. 732–737, doi:10.1109/GRC.2006.1635905.
[36] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, Smote: synthetic minority over-sampling technique, J. Artif. Intell. Res. 16 (1) (2002) 321–357, doi:10.1613/jair.953.
[37] H. Han, W.-Y. Wang, B.-H. Mao, Borderline-smote: a new over-sampling method in imbalanced data sets learning, in: Proceedings of the 2005 International Conference on Advances in Intelligent Computing, volume 1, Springer, Hefei, China, 2005, pp. 878–887, doi:10.1007/11538059_91.
[38] J. Luengo, A. Fernández, S. García, F. Herrera, Addressing data complexity for imbalanced data sets: analysis of smote-based oversampling and evolutionary undersampling, Soft Comput. 15 (10) (2011) 1909–1936, doi:10.1007/s00500-010-0625-8.
[39] N. Qazi, K. Raza, Effect of feature selection, synthetic minority over-sampling (SMOTE) and under-sampling on class imbalance classification, in: 2012 14th International Conference on Modelling and Simulation, IEEE, Cambridge, UK, 2012, pp. 145–150, doi:10.1109/UKSim.2012.116.
[40] A. Tesfahun, D.L. Bhaskari, Intrusion detection using random forests classifier with smote and feature reduction, in: 2013 International Conference on Cloud & Ubiquitous Computing & Emerging Technologies, IEEE, Pune, India, 2013, pp. 127–132, doi:10.1109/CUBE.2013.31.
[41] Y. Hong, U. Hwang, J. Yoo, S. Yoon, How generative adversarial networks and their variants work: an overview, ACM Comput. Surv. (CSUR) 52 (1) (2019) 10, doi:10.1145/3301282.
[42] J. Lin, Divergence measures based on the Shannon entropy, IEEE Trans. Inf. Theory 37 (1) (1991) 145–151, doi:10.1109/18.61115.
[43] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, D.N. Metaxas, StackGAN: text to photo-realistic image synthesis with stacked generative adversarial networks, in: Proceedings of the IEEE International Conference on Computer Vision, IEEE, Venice, Italy, 2017, pp. 5907–5915.
[44] L. Yu, W. Zhang, J. Wang, Y. Yu, SeqGAN: Sequence generative adversarial nets with policy gradient, in: Thirty-First AAAI Conference on Artificial Intelligence, AAAI, San Francisco, California, USA, 2017, pp. 1–7.
[45] K. Lei, M. Qin, B. Bai, G. Zhang, M. Yang, GCN-GAN: a non-linear temporal link prediction model for weighted dynamic networks, in: IEEE INFOCOM 2019-IEEE Conference on Computer Communications, IEEE, Paris, France, France, 2019, pp. 388–396, doi:10.1109/INFOCOM.2019.8737631.
[46] H. Shi, J. Dong, W. Wang, Y. Qian, X. Zhang, SSGAN: secure steganography based on generative adversarial networks, in: 18th Pacific-Rim Conference on Multimedia, volume 1, Springer, Harbin, China, 2017, pp. 534–544, doi:10.1007/978-3-319-77380-3_51.
[47] G. Douzas, F. Bacao, Effective data generation for imbalanced learning using conditional generative adversarial networks, Expert Syst. Appl. 91 (2018) 464–471, doi:10.1016/j.eswa.2017.09.030.
[48] L. Vu, C.T. Bui, Q.U. Nguyen, A deep learning based method for handling imbalanced problem in network traffic classification, in: Proceedings of the Eighth International Symposium on Information and Communication Technology, in: SoICT 2017, ACM, Nha Trang City, Viet Nam, 2017, pp. 333–339, doi:10.1145/3155133.3155175.
[49] S. Kullback, R.A. Leibler, On information and sufficiency, Ann. Math. Stat. 22 (1) (1951) 79–86, doi:10.1214/aoms/1177729694.
[50] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, arXiv preprint arXiv:1412.6980(2014) 1–15.
[51] M. Tavallaee, E. Bagheri, W. Lu, A.A. Ghorbani, A detailed analysis of the KDD cup 99 data set, in: 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, IEEE, Ottawa, ON, Canada, 2009, pp. 1–6, doi:10.1109/CISDA.2009.5356528.
[52] N. Moustafa, J. Slay, 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), IEEE, Canberra, ACT, Australia, 2015, pp. 1–6, doi:10.1109/MilCIS.2015.7348942.
[53] J. Cervantes, X. Li, W. Yu, K. Li, Support vector machine classification for large data sets via minimum enclosing ball clustering, Neurocomputing 71 (4–6) (2008) 611–619, doi:10.1016/j.neucom.2007.07.028.
[54] L. Bertinetto, J. Valmadre, J.F. Henriques, A. Vedaldi, P.H. Torr, Fully-convolutional siamese networks for object tracking, in: European Conference on Computer Vision, Springer, Amsterdam, Netherlands, 2016, pp. 850–865, doi:10.1007/978-3-319-48881-3_56.
[55] V. Mnih, K. Kavukcuoglu, D. Silver, A.A. Rusu, J. Veness, M.G. Bellemare, A. Graves, M. Riedmiller, A.K. Fidjeland, G. Ostrovski, et al., Human-level control through deep reinforcement learning, Nature 518 (7540) (2015) 529–533, doi:10.1038/nature14236.
[56] R.C. Daudt, B. Le Saux, A. Boulch, Fully convolutional siamese networks for change detection, in: 2018 25th IEEE International Conference on Image Processing (ICIP), IEEE, Athens, Greece, 2018, pp. 4063–4067, doi:10.1109/ICIP.2018.8451652.
[57] V. Mnih, N. Heess, A. Graves, et al., Recurrent models of visual attention, in: Advances in Neural Information Processing Systems, Montral, Canada, 2014, pp. 2204–2212.

Shuokang Huang received the B.S. degree in Electronic Information Science and Technology from Sun Yat-sen University, Guangzhou, Guangdong, China, in 2018. He is currently working toward the M.S. degree in Peking University. His currently research interest is Future Internet Architecture and Anomaly Detection.

Kai Lei is an Associate Professor of Peking University, Beijing, China. He received the M.S. degree in Computer Science from Columbia University, New York, USA, in 1999, and Ph.D. degrees in Computer Science from Peking University, Beijing, China, in 2015. His currently research interest is Future Internet Architecture (ICN/NDN, IoT), Blockchain, Federal Learning and Knowledge Graph.