Effective Intrusion Detection in Highly Imbalanced IoT Networks With Lightweight S2CGAN-IDS

Caihong Wang, Du Xu, Zonghang Li, and Dusit Niyato, *Fellow, IEEE*

***Abstract*—Since the advent of the Internet of Things (IoT), exchanging vast amounts of information has increased the number of security threats in networks. As a result, intrusion detection based on deep learning (DL) has been developed to achieve high throughput and high precision. Unlike general DL-based scenarios, IoT networks contain benign traffic far more than abnormal traffic, with some rare attacks. However, most existing studies have been focused on sacrificing the detection rate of the majority class in order to improve the detection rate of the minority class in class-imbalanced IoT networks. Although this way can reduce the false negative rate of minority classes, it both wastes resources and reduces the credibility of the intrusion detection systems. To address this issue, we propose a lightweight framework named S2CGAN-IDS. The proposed framework leverages the distribution characteristics of network traffic to expand the number of minority categories in both data space and feature space, resulting in a substantial increase in the detection rate of minority categories while simultaneously ensuring the detection precision of majority categories. To reduce the impact of sparsity on the experiments, the CICIDS2017 numeric data set is utilized to demonstrate the effectiveness of the proposed method. The experimental results indicate that our proposed approach outperforms the superior method in both Precision and Recall, particularly with a 10.2% improvement in the F1-score.**

***Index Terms*—Class imbalance, deep learning (DL), generative adversarial networks, Internet of Things (IoT), intrusion detection.**

I. Introduction

THE EMERGENCE of the 5G era has brought new challenges to cybersecurity due to the proliferation of the Internet of Things (IoT). IoT devices are known to harbor a significant amount of private information and are often secured with simple encryption. As a result, a considerable number of these devices may be rendered as zombie hosts or utilized as mining tools, with some users even falling prey to cyber extortionists.

Intrusion detection systems (IDSs) [1] constitute a pivotal component of firewalls and can detect viruses before they reach IoT devices. As such, IDS has become an indispensable preventive measure for ensuring the security of IoT networks. Conventional intrusion detection technologies heavily rely on manually crafted rules and signatures. The creation and upkeep of these rules and signatures require significant time and labor. However, the contemporary proliferation of traffic and the rising prevalence of attacks stemming from the IoT have rendered these traditional methods relatively ineffective.

To overcome the limitations of traditional intrusion detection methods, deep learning (DL) has surfaced as a promising approach. DL algorithms, including deep belief networks (DBNs) [2], convolutional neural networks (CNNs) [3], and recurrent neural networks (RNNs) [4], can automatically learn complex patterns and anomalies from raw network traffic. This enables more accurate automatic detection of potential threats. Additionally, DL algorithms can effectively leverage massive network traffic to identify potential attacks and adapt to changing attack patterns, thereby significantly enhancing the detection accuracy of the IDS.

The DL algorithm has exhibited high accuracy in detecting network attacks [5]. However, to operate effectively, DL models require an adequate number of training examples. In comparison to the abundance of benign network traffic examples, certain attack categories are scarce in number. Consequently, DL-based intrusion detection models encounter the challenge of a high false-negative rate [6], [7], [8].

In the realm of class imbalance, researchers have endeavored to optimize the efficiency of DL techniques, including data-level approaches [9], [10], algorithm-level strategies [11], [12], integrated learning [13], [14], transfer learning [15], [16], and evaluation metrics [17], with the primary aim of mitigating the false-negative rate of IDSs. However, this objective often comes at the expense of precision for majority classes, while improving the detection rate of minority attacks. Therefore, these methods may ultimately not only compromise the reliability of the system but also waste resources.

The motivation of this article is to enhance the detection rate of minority categories in IoT networks while minimizing the impact on the detection rate of majority categories. By focusing on the distinctive characteristics of attack frequency, we try to pay more attention to the extremely rare attacks and foster the advancement and innovation of this field from different angles.

In response to the aforementioned concerns, we present a proficient and lightweight S2CGAN-IDS framework that leverages the distribution characteristics of traffic categories within IoT networks. Our framework extends the original

Manuscript received 6 June 2023; revised 16 August 2023; accepted 10 December 2023. Date of publication 13 December 2023; date of current version 25 April 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62171085. *(Corresponding author: Du Xu.)*

Caihong Wang, Du Xu, and Zonghang Li are with the School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China (e-mail: xudu@uestc.edu.cn).

Dusit Niyato is with the School of Computer Science and Engineering, Nanyang Technological University, Singapore.

Digital Object Identifier 10.1109/JIOT.2023.3342638

imbalanced training data by considering two distinct perspectives: 1) data space and 2) feature space. This approach aims to enhance the detection rate of underrepresented categories while maintaining satisfactory detection rates for the majority classes.

The main contributions of this article are summarized as follows.

1) We have devised a lightweight S2CGAN-IDS framework from a data-oriented perspective to address the issue of class imbalance. This framework aims to improve the detection rate of the underrepresented minority class while maintaining accuracy for the majority class.
2) This article presents an innovative feature extraction method that combines Siamese networks and autoencoders to preserve class differences and which leads to stable training of the adversarial generative networks.
3) This article presents a novel data augmentation technique named SCGAN which utilizes category-similar features as auxiliary information for the generator, effectively steering the generator toward a faster approximation of the authentic data distribution.
4) This article introduces a highly efficient data synthesis approach named synthetic k neighbors (SKNs) that utilizes feature space-based methods to generate samples for categories that are extremely rare.

II. Related Works

Intrusion detection systems are an essential part of IoT network security, playing a crucial role in detecting malicious network activities. With the proliferation of IoT devices and the increasing sophistication of attacks, there is a growing need for robust and efficient IDS. In recent years, DL has emerged as a promising approach to enhance the effectiveness of IDS. In this section, we provide a concise overview of the literature on DL-based intrusion detection, with a specific focus on addressing the problem of class imbalance in deep intrusion detection systems.

A. *Deep Learning-Based IDS*

The utilization of DL has gained substantial popularity in the domain of intrusion detection. Various DL techniques, including DBNs [18], [19], CNNs [20], [21], [22], and RNNs [23], [24], have been employed for this purpose.

DL has demonstrated its efficacy in detecting abnormal patterns in network traffic and identifying potential intrusions. However, in order to achieve optimal performance, DL models necessitate comprehensive data distribution. Regrettably, intrusion detection systems often encounter the challenge of class imbalance, leading to inadequate performance of DL models, particularly in detecting attacks that belong to the minority class.

B. *Class Imbalance in IDS*

The issue of class imbalance is prevalent in intrusion detection, where the number of normal samples outweighs the number of attack samples. To tackle this challenge, researchers have proposed various techniques from three different perspectives: 1) the loss function or structure of the classifier; 2) the synthesis of minority class samples from the feature space; and 3) the generation of new minority class samples from the data space.

In terms of the classifier, Wang et al. [25] proposed a new loss function called mean false error (MFE) along with its improved version, mean squared false error (MSFE), that captures errors of both majority and minority classes equally, providing a solution to the data imbalance problem in deep networks from an evaluation perspective. Bedi et al. [26], [27] employed Siamese neural networks in Siam-IDS and its improved variant, I-SiamIDS, to achieve higher recall values for both R2L and U2R attack classes. Meanwhile, Gupta et al. [28], [29] proposed two intrusion detection methods: 1) CSE-IDS, which integrates the extreme gradient boosting (XGBoost) algorithm with sensitive unbalanced deep intrusion detection and 2) LIO-IDS, based on long short-term memory (LSTM) and One-versus-One algorithms, both of which demonstrate high detection rates and reduced computational costs.

However, these approaches often come at the expense of sacrificing the detection accuracy of majority categories, resulting in an increased detection rate for minority categories. This tradeoff undermines the reliability of intrusion detection systems.

For the method of synthesizing minority classes from the feature space, Al and Dener [30] developed the CNN-LSTM method, which combines a hybrid deep learning (HDL) approach with STL (SMOTE + Tomek-Link) class imbalance processing to enhance intrusion detection performance. Hasib et al. [31] proposed a hybrid method that combines KNN undersampling with SMOTE oversampling to enhance the data set of network intrusion detection systems.

Merely synthesizing new samples of minority categories from the feature space alone is insufficient to capture the comprehensive information present in the high-dimensional space of minority category attacks. Consequently, the detection rate of these methods for minority categories remains severely limited.

Fortunately, the emergence of generative adversarial network (GAN) has brought about new techniques for generating new samples of minority classes from the data space. These advancements bring renewed hope for effectively addressing the class imbalance problem in IDS.

Lee and Park [32] proposed the use of GAN to generate synthetic data to balance data sets and improve the performance of imbalanced IDS. Recent studies have shown that GANs can effectively generate realistic attack traffic [33] and enhance the accuracy of intrusion detection models [34]. Huang and Lei [35] proposed imbalance generative adversarial networks (IGANs) to generate representative samples for minority classes, while Cui et al. [36] introduced a Wasserstein GAN (WGAN) module to address unbalanced data.

These approaches have shown notable advancements in addressing the class imbalance between minority and majority classes compared to generating synthetic data solely from the feature spaces. However, in the domain of wireless networks,

TABLE I
SOME DETAILS OF COMMONLY USED CLASSIC DATA SETS

Dataset	Year	Characteristics	Sparsity	Frequency
NSLKDD	1999	Network-based, real-world traffic, KDD Cup 1999	Medium	Uniform
UNSW-NB15	2015	Network-based, real-world traffic, contains synthetic and real data	High	Gradual
CICIDS2017	2017	Network-based, real-world traffic, contains IoT and normal traffic	Low	Stepped

there are highly rare attacks that can inflict significant damage on network devices if they occur. Regrettably, the aforementioned method of generating new attacks from the data space is not effective in addressing these types of highly rare attacks. In such scenarios, alternative approaches need to be explored to tackle the challenges posed by these exceptionally rare attack instances.

III. MOTIVATION

The class imbalance problem in IoT scenarios is of paramount importance in ensuring IoT security [37]. This problem stems from several key factors, including the extensive deployment of devices, the wide variety of malicious behaviors, the limited resources of IoT devices, and the heightened sensitivity of security requirements. These factors collectively contribute to the scarcity of malicious behavior data in IoT scenarios, making accurate detection of such behaviors an urgent necessity [38]. Consequently, the effective resolution of the class imbalance problem holds significant significance in upholding IoT security.

After an extensive literature search, NSLKDD, UNSW-NB15, and CICIDS2017 have emerged as the predominant data sets utilized in this field over the past two decades. An evaluation of attack frequency across these data sets reveals a distinct gradient shift. Notably, CICIDS2017 exhibits a conspicuous step-like upgrade while possessing the most recent and sparsest characteristics, aligning it more closely with the traffic observed in real IoT network environments [39]. Consequently, CICIDS2017 has been chosen for subsequent analysis and experimentation.

Based on the analysis mentioned above, the fundamental issue that must be addressed by an effective IoT network intrusion detection model is enhancing the detection rate of the minority categories while maintaining the accuracy of the majority categories. To tackle this problem, we conducted principal component analysis (PCA) on a widely used intrusion detection data set and generated a scatterplot based on the resulting PCA data (Fig. 1).

Analysis of Fig. 1 reveals that categories located in the upper-left region of the scatterplot possess an ample number of samples and exhibit a complete distribution. Conversely, the categories in the middle region are relatively scarce, and the distribution outline is rather rough. Finally, categories situated in the lower-right region contain only a few, scattered data points.

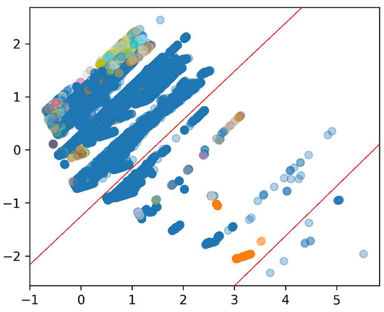

Fig. 1. Scatter of CICIDS2017 data set.

TABLE II
LABELS, IR$_i$ VALUES, AND QUANTITY LEVELS OF THE CICIDS2017 DATA SET

Subclass	Label	IR_i	Level
BENIGN	BENIGN		ample
DoS DoS Hulk DDoS DoS GoldenEye DoS slowloris DoS Slowhttptest	DoS/DDoS	5.98	
PortScan	PortScan	14.31	
FTP-Patator SSH-Patator	Patator	164.33	scarce
Web Attack-Brute Force Web Attack-XSS Web Attack-Sql Injection	Web Attack	1042.28	
Bot	Bot	1156.92	
Infiltration	Infiltration	63141.03	rare
Heartbleed	Heartbleed	206645.18	

Based on the observed characteristics in the scatter, we calculate the imbalance ratio (IR$_i$) for each class by $n_{\max}/n_i$, where n_i represents the number of the *type-i* attack, and $n_{\max}$ represents the number of normal samples. Remarkably, our calculations revealed a distinct step-wise distribution pattern in the IR$_i$ values, which aligned with the visual representation depicted in the scatter.

By considering the step distribution of IR$_i$ values and its coherence with the visual depiction of the scatter, we classify intrusion detection traffic into ample-level, scarce-level, and rare-level (as shown in Table II), and treat them differently based on their respective attributes.

To minimize the computational overhead while ensuring a high detection rate for the majority category, we specifically avoid processing the majority category (ample-level), which already exhibits a complete distribution.

In the scenario addressed in this article, a challenge arises due to the significant disparity in the number of minority samples. Solely relying on data space-based data augmentation methods to generate minority samples may be ineffective for rare-level categories, as depicted in the lower right part of Fig. 1, where the scarcity of samples hinders the generation of new instances. Conversely, employing only feature space-based data enhancement methods to synthesize minority samples may result in synthetic samples that closely resemble the original ones. Consequently, the performance of the scarce-level categories in the middle part of Fig. 1 may be constrained.

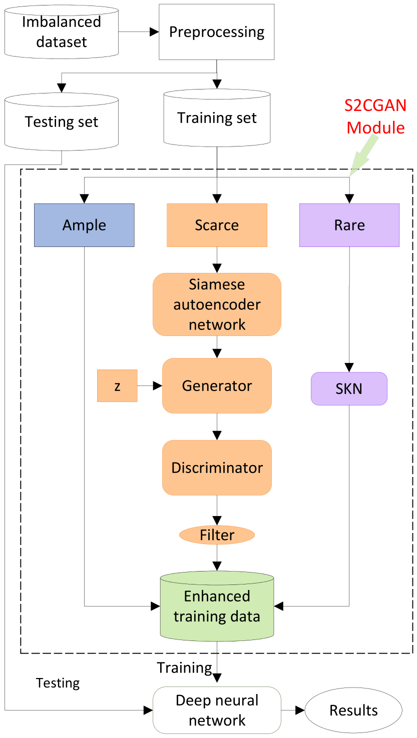

Fig. 2. S2CGAN-IDS model framework. The framework diagram for the proposed algorithm consists of three primary components: data preprocessing, data enhancement, and training and testing of the IDS classifier.

Consequently, we partition the minority categories into scarce-level and rare-level. For scarce-level categories, we adopt advanced data space-based data augmentation methods, while for rare-level categories, we rely on feature space-based data enhancement techniques.

The remaining parts of this article are organized as follows. Section IV gives the outline of our framework. Section V introduces the main algorithms used to design the proposed S2CGAN. Section VI presents a detailed explanation of the architecture and results of the experiments conducted in this study. The conclusion and challenges for future work are given in Section VII.

IV. Methodology Overview

In this section, we present our lightweight intrusion detection framework, which comprises three primary components as shown in Fig. 2.

These include data set processing, the S2CGAN module, and classifier training and testing. Our framework is designed to improve the performance of intrusion detection systems in highly imbalanced data sets by employing different data augmentation techniques for different category levels. The specific detection process of the IDS framework is as follows.

Data Set Processing: The data set is processed through the following steps: normalization and train-test split.

S2CGAN Module: As a case study, we classify all categories within this data set into three levels based on the step-change characteristics of their respective numbers. And employ the S2CGAN module to enhance the data set, which is a data generation model incorporating two techniques: 1) SCGAN and 2) SKN. The SCGAN is utilized to generate scarce-level attacks, while a filter is applied to the generated data to enhance the consistency of generated samples and original samples. On the other hand, SKN is used to generate rare-level attacks by simulating potential rare-level attack distributions through the KNN algorithm.

Algorithm 1: S2CGAN (Main)

```
Data: Original training dataset $T_o$;
      The SCGAN threshold $\eta$.
Result: Argument training dataset $T_a$.
Initialize $T_a$ as an empty dataset;
Calculate IR$_i$ values of each class and divide them into ample-level, scarce-level, and rare-level based IR$_i$ rates from low to high;
if ample-level categories then
    Add ample-level samples into $T_a$;
else if scarce-level categories then
    Pretrain the SAE model by using Algorithm 2;
    for each epoch do
        Input the encoder result of SAE $s$ and random noise $z$ to conditional generative adversarial network (CGAN);
        Calculate the losses of the generator $G$ and the discriminator $D$;
        Update the generator $G$ and the discriminator $D$ with their losses;
    end
    Use the generator $G$ of well-trained SCGAN to generate new scarce-level samples $G(z|s)$;
    for each $G(z|s)$ do
        Input $G(z|s)$ into the discriminator $D$ and output $(D(G(z|s))$;
        if $(D(G(z|s)) \geq \eta)$ then
            Add $G(z|s)$ and its label into $T_a$;

    end
    Add scarce-level samples into $T_a$;
else if rare-level categories then
    Synthesize rare-level samples with Algorithm 3;
    Add the synthesized samples and original rare-level samples into $T_a$;
return The augmented training dataset $T_a$;
```

Training/Testing: To evaluate the effectiveness of the proposed data augmentation algorithm, we implemented it using a deep neural network (DNN) and trained an intrusion detection classifier using the augmented data set. The performance of the resulting classifier was then verified using a separate test set.

The S2CGAN module plays a central role in the algorithm proposed in this article. This module enhances the original data set by operating in both the data and feature spaces, thus improving the detection accuracy of minority categories in the intrusion detection classifier. In the subsequent sections of this article, we will present a detailed exposition of the S2CGAN (Algorithm 1).

Algorithm 2: SAN

```
Input: Training dataset $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$.
Output: Model parameters $W_s$, $W_d$, $b_s$, $b_d$.
Initialize Encoder parameters $W_s$, $b_s$;
Initialize Decoder parameters $W_d$, $b_d$;
for each epoch do
    Randomly select two different samples $(x_i, y_i)$ and $(x_j, y_j)$;
    Feed each sample through a shared encoder to obtain their encoder outputs $f(x_i)$ and $f(x_j)$;
    Use $f(x_i)$ and $f(x_j)$ as inputs to a shared decoder, producing the output $\bar{x}_i$ and $\bar{x}_j$;
    Compute the loss function and backpropagate it through the network;
    Update model parameters $W_s$, $W_d$, $b_s$, $b_d$;
end
return model parameters $W_s$, $W_d$, $b_s$, $b_d$;
```

V. Implementation Details of S2CGAN

As demonstrated in the PCA scatter plot of the CICIDS2017 data set in Fig. 1, the data distributions for ample-level categories are complete and can be directly reserved in the augmented data set. However, scarce-level attacks have only approximate distribution. In this situation, we utilize SCGAN to learn the original distribution and generate missing data. Finally, we employ SKN to expand the original distribution as much as possible from the feature space for rare-level attacks with only a few data points.

A. SCGAN for Scarce-Level Attacks

This section is dedicated to the details of the SCGAN module, which serves the purpose of generating scarce-level categories. The SCGAN module consists of a Siamese autoencoder network (SAN) and a GAN. First, we introduce the SAN model to extract differential feature information for the SCGAN module. Algorithm 2 shows the detail.

SAN comprises a pair of autoencoders (AEs) with Siamese neural networks (SNNs). The SNNs are designed to capture the differences between the extracted key features of various attacks. While the AEs are responsible for extracting the most significant features. By integrating both models in the SAN, the features can be extracted more effectively and efficiently, resulting in the creation of scarce-level attacks. Fig. 3 provides a detailed illustration of the SAN.

SNN is a type of neural network that consists of two identical subnetworks, each taking a different input (i.e., x_1 and x_2) but with the same architecture, parameters, and weights. The two subnetworks output a pair of feature vectors $(f_w(x_1), f_w(x_2))$, which can be used to compute the similarity or difference between the two inputs. In this article, the Euclidean distance (1) was used to calculate the similarity between the two feature vectors

$$E_w(x_1, x_2) = ||f_w(x_1) - f_w(x_2)||. \tag{1}$$

The parameters can be optimized by minimizing the reconstruction error, which is typically computed with the following formula:

$$\mathcal{L}_{\text{SNN}} = \frac{1}{2}\sum\Big((1-y)E_w^2 + y\max(0, m - E_w)^2\Big) \tag{2}$$

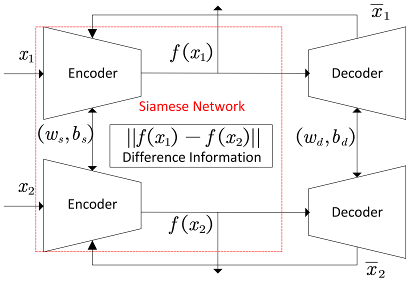

Fig. 3. Siamese AE networks. It consists of a pair of encoders sharing parameters and a pair of decoders sharing parameters.

where y is the binary label indicating whether the input pairs (x_1, x_2) are similar $(y = 0)$ or dissimilar $(y = 1)$. E_w^2 in the loss function computes the squared distance between the feature vectors of similar pairs, while $\max(0, m - E_w)^2$ computes the squared distance between the feature vectors of dissimilar pairs, with a margin m to ensure that the distance is smaller than m.

AE is a powerful unsupervised learning algorithm that is widely used in the field of machine learning. Unlike supervised learning methods, AE does not rely on labeled data for training. It consists of two critical components: 1) encoder and 2) decoder. The encoder is responsible for extracting the essential features $f_w(x)$ of the original data x, and the decoder aims to reconstruct the input data based on these extracted features. By minimizing the error between the reconstructed data $\bar{x}$ and the original input data during the learning process, AE learns the implicit feature representation of the data. In this study, the cost function is given by

$$\mathcal{L}_{\text{AE}}(x, \bar{x}) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x}_i)^2. \tag{3}$$

GAN is employed to learn the underlying data distribution and generate novel samples to fill in the missing data. It was first introduced by Goodfellow et al. [40] as a powerful approach for generating realistic images by training a generator network to produce images that are indistinguishable from real images by a discriminator network. Initially, GAN was used for generating realistic images from random noise, but they have since been extended to other domains, such as text and music generation. In addition to generating new data, GAN can also be used for other tasks, such as data augmentation, where the generator is used to produce additional training data for a given task.

However, the original GAN generates samples that belong to a wide range of classes, without any control over the specific class of the generated samples. To overcome this limitation, CGAN [41] was introduced in the same year as an extension of GAN. CGAN incorporates conditional information into the generator G and the discriminator D outputs the probability of real and fake samples. This enables the generation of samples based on specific input conditions, allowing for targeted data synthesis.

In the realm of network intrusion detection, CGAN typically exploits labels as conditional information to ensure that the generated samples correspond to specific predefined categories. Nevertheless, CGANs frequently encounter substantial challenges, prominently characterized by training instability and a phenomenon known as mode collapse [42], [43], [44], [45].

To effectively address the potential deterioration in the quality of generated samples attributable to the aforementioned challenges, our proposed SCGAN integrates more informative conditional information. The SAN meticulously extracts and processes conditional information, enhancing the generator's capacity to discern underlying distribution patterns. This strategic incorporation of enriched contextual cues amplifies the generator's efficacy, effectively steering the generator toward a faster approximation of the authentic data distribution and consequently bolstering GAN stability [46], [47], [48].

To achieve this goal, we employ difference information s and random noise $z \in N(\mu, \sigma^2)$ obtained from the SAN model as input for CGAN. The generator G takes s and z as the input and produces a set of pseudo samples $G(z|s)$, while the discriminator D takes both the real attack samples x and generated attack samples $G(z|s)$ as input. The fundamental objective of SCGAN is demonstrated through $\min_G \max_D \mathcal{V}(D, G)$, as in

$$\mathcal{V}(D, G) = E_{x \sim p_x} \log(D(x)) + E_{z \sim p_z} \log(1 - D(G(z|s))). \quad (4)$$

To train the discriminator D, we feed the conditional information s and the noise vector z into the generator G, which generates the pseudo samples $G(z|s)$. Subsequently, we feed both the real samples x and the generated samples $G(z|s)$ into D and update the parameters of D based on the loss function $\mathcal{L}_D$ of the discriminator as shown in the following:

$$\mathcal{L}_D = -\log(D(x)) - \log(1 - D(G(z|s))). \quad (5)$$

To train the generator G, we fix the parameters of the discriminator D and the pseudo samples $G(z|s)$ into D. The error backpropagates to G and its parameters are updated based on the loss function $\mathcal{L}_G$ to enhance its performance

$$\mathcal{L}_G = -\log(D(G(z|s))). \quad (6)$$

The alternating iterative training process of the SCGAN model persists until it reaches the stationary local Nash equilibrium [49], wherein the discriminator is unable to effectively differentiate between the pseudo samples and the real samples. At this point, both the generator and the discriminator have been effectively trained, enabling the SCGAN model to generate high-quality pseudo samples that exhibit a close resemblance to the actual attack samples.

After the GAN converges, the generated samples by the generator closely resemble real scarce-level attacks. However, it is possible for some noise points to still be present in the generated samples. To address this concern, a filter is applied after the discriminator to eliminate improperly generated samples. This ensures that the training of the intrusion detection classifier remains unaffected by the additional interference introduced by these samples.

Algorithm 3: Synthesis K Neighbors (SKN)

```
Input: Training dataset $T_o$, where $X = x_1, x_2, \ldots, x_n$;
        Class distribution $C$, where $C = c_1, c_2, \ldots, c_l$;
        The number of class $i$ that need to be synthesized $N_i$;
        K-nearest neighbors parameter $k$.
Output: Synthetically over-sampled dataset $T_r$.
Initialize $T_r$ as an empty dataset;
for each rare-level class $c_i$ do
    Find the k-nearest neighbors of each rare-level class sample $x_i$;
    for each sample $x_i$ do
        Choose $N_i$ k-nearest neighbors randomly and denote them as $x_1, x_2, \ldots, x_{N_i}$;
        for each $j = 1, 2, \ldots, N_i$ do
            Choose a random number $\lambda \in [0, 1]$;
            Set $x_{new} = x_i + \lambda \times (x_j - x_i)$;
        end
        Add the new sample $x_{new}$ and $c_i$ into $T_r$;
    end
end
return the over-sampled dataset $T_r$;
```

B. SKN for Rare-Level Attacks

Rare-level attacks are differentiated from scarce-level attacks by their characteristics of having only a few scattered points. The sparsity of data at this level presents a challenge for deep neural networks to accurately simulate the possible distributions. To overcome this issue, we introduce a feature space-based oversampling method called SKN. This method specifically addresses the limited number of samples at this level, effectively avoiding the challenges associated with insufficient data.

Oversampling is a widely adopted technique to tackle imbalanced data sets, and popular methods include random over sampling (ROS), SMOTE, and its variants. ROS involves replicating minority class instances to balance the class distribution while retaining the original information. However, it is susceptible to overfitting since it duplicates the same information multiple times. In contrast, SMOTE employs the KNN algorithm in the feature space to generate new samples and equalize the data set. Nevertheless, SMOTE may encounter challenges related to high redundancy, resulting in an augmented computational burden and potentially compromising generalization capability [50].

In this study, we propose a lightful rare-level oversampling method called SKN (Algorithm 3). SKN employs SMOTE as a basic algorithm for the generation of synthetic samples, accomplished through the implementation of elementary feature interpolation connecting target samples from the rare-level attacks and their proximate neighbors. This process is intended to augment the rare-level categories equilibrium. In contrast to the conventional SMOTE technique, SKN places a deliberate emphasis on the streamlining of methodology and the optimization of computational efficiency. Thus, SKN demonstrates particular pertinence in contexts characterized by limitations in available resources.

The method involves three key steps. First, we identify k adjacent samples of the ith rare class by means of a calculation process. Second, we randomly select N_i samples from the adjacent set. Finally, we randomly apply a transformation function to the selected samples as well as the rare class samples themselves in order to synthesize new samples.

VI. Experiment

A. Benchmark Data Set

Our analysis of commonly used data sets from 1998 has revealed that most of them are outdated and unreliable. Some of these data sets suffer from limited traffic diversity and capacity, while others lack coverage of various known attacks. Additionally, some anonymize packet payload data, which makes them unable to reflect current trends. Moreover, several data sets lack essential features and metadata, which are necessary for accurate analysis.

For our research, we utilized the CICIDS2017 data set, which is a publicly available data set composed of network traffic collected from the Canadian Institute for Cybersecurity (CIC) during weekdays in 2017. This data set includes 50 subcategories and seven major types of attacks and has gained popularity as a replacement for the previously widely used KDDCup99 and NSLKDD data sets due to its realistic and diverse traffic scenarios. The CICIDS2017 [51] data set provides a reliable resource for studying network intrusion detection methods and has been widely adopted by researchers and practitioners in the field.

The CICIDS2017 data set [51] consists of benign and recent common attacks. The data set collection spanned five days and concluded on 7 July 2017, at 5:00 P.M. The attacks were executed on Tuesday, Wednesday, Thursday, and Friday morning and afternoon, respectively.

CICFlowMeter was utilized as a flow feature extraction tool to generate a CSV file with over 80 features based on the submitted PCAP file, which contains the flow data of the network interface card that can be obtained via Wireshark software or flow sniff function. There are two modes: 1) online and 2) offline. The online mode allows for real-time monitoring and feature generation, which can be saved locally after monitoring, whereas the offline mode entails the submission of a PCAP file and the receipt of a CSV file containing features.

After analyzing the data set, we observed that some feature values exhibited significant variation. This wide disparity in feature values has the potential to result in slow network convergence and neuron output saturation. Therefore, we deemed it necessary to normalize the data set. In this study, we employed the Min–Max normalization method, which resulted in the normalization of the data to a range of [0, 1]. This approach enhances the comparability and compatibility of the features, mitigating the effects of the initial variation in the data set and improving the overall quality of the results. The equation for the Min–Max normalization is as follows:

$$x_{\text{norm}} = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{7}$$

TABLE III
Detailed Size of the Training Set, Test Set, and Augmented Data Sets

Category	Training(80%)	Testing(20%)	After Augmentation
BENIGN	1818476	454620	1818476
DoS/DDoS	304550	76138	304550
PortScan	127144	31785	127144
Patator	11068	2767	127144
Web Attack	1744	436	127144
Bot	1573	393	127144
Infiltration	29	7	1573
Heartbleed	9	2	1573

where x_{norm} is the normalized value, x is the original feature value, $x_{\min}$ is the minimum value of the feature, and $x_{\max}$ is the maximum value of the feature.

In accordance with best practices in machine learning, we partitioned the preprocessed data set into training and test sets, with an 8:2 ratio, respectively. It is important to note that the test set was only used during the evaluation phase of the IDS to ensure the validity and reliability of our experimental methodology.

We analyzed the imbalance ratio (IR_i) for all categories in the training set and observed a stepwise distribution of the imbalance ratio value. Notably, we identified a significant gap between the first and third echelons. To address this, we employed a categorization scheme where we grouped *DoS/DDoS* and *PortScan* attacks into the *ample* level, while *Patator*, *Web Attack*, and *Bot* attacks were assigned to the *scarce* level. *Infiltration* and *Heartbleed* attacks were categorized as *rare* level. Table II shows the specific division of labels in the CICIDS2017 data set and the corresponding attack levels according to the IR_i team.

In this experiment, the CICIDS2017 data set is randomly partitioned into training and test sets to validate the effectiveness of our approach. The training set is utilized to train the S2CGAN model and generate a sufficient number of samples as outlined in Table III. Subsequently, the IDS classifier is trained using the augmented data set. To ensure unbiased and fair results, the test set is exclusively used to evaluate the performance of the IDS classifier.

B. Evaluation Metrics

To comprehensively assess the performance of the proposed IDS in this article, we employ Precision, Recall, and F-Score as evaluation metrics. These metrics are calculated at the subcategory level, using both weighted and macro averages. In this context, TP represents true positives, TN represents true negatives, FP represents false positives, and FN represents false negatives. These metrics provide a comprehensive evaluation of the system's performance in terms of accuracy, sensitivity, and overall effectiveness in detecting intrusions.

Precision is a metric that measures the proportion of correctly identified positive samples to all samples that were predicted as positive. Mathematically, it can be represented as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{8}$$

Recall, also known as sensitivity or true positive rate, represents the number of samples of a specific class that were correctly identified out of all the samples belonging to that class. It can be calculated using the following equation:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{9}$$

The F-Score is a widely used metric that balances both Precision and Recall and provides a more comprehensive evaluation of the performance of a classifier. It is defined as the harmonic mean of Precision and Recall, and is given by the following formula:

$$F_\beta = \frac{\left(1+\beta^2\right)\text{Precision} \times \text{Recall}}{\beta^2(\text{Precision} \times \text{Recall})}. \tag{10}$$

The coefficient β is used to describe the relative importance of Precision and Recall. For this particular experiment, we set β as 1, which refers to the F1-score.

For the highly imbalanced data set used in this experiment, using the weighted average formulas in (11) for the overall indicator may introduce a bias toward ample-level categories. This bias occurs due to the vast quantity of samples belonging to the ample-level categories compared to the minority categories. As a result, the weighted average may overly prioritize the performance of the ample-level categories and may not accurately reflect the performance of the minority-level categories

$$\begin{aligned} P_{\text{weighted}} &= \frac{\sum_{i=1}^{n} w_i \times P_i}{\sum_{i=1}^{n} w_i} \\ R_{\text{weighted}} &= \frac{\sum_{i=1}^{n} w_i \times R_i}{\sum_{i=1}^{n} w_i} \\ F_{1\,\text{weighted}} &= \frac{\sum_{i=1}^{n} w_i \times F_{1i}}{\sum_{i=1}^{n} w_i}. \end{aligned} \tag{11}$$

Here, n is the total number of samples, P_i, R_i, and F_{1i} are the Precision, Recall, and F1-score values for class i, respectively, and w_i is the weight assigned to class i, which is proportional to the frequency of that class in the data set

$$\begin{aligned} P_{\text{macro}} &= \frac{1}{n}\sum_{i=1}^{n} P_i \\ R_{\text{macro}} &= \frac{1}{n}\sum_{i=1}^{n} R_i \\ F_{1\,\text{macro}} &= \frac{1}{n}\sum_{i=1}^{n} F_{1i}. \end{aligned} \tag{12}$$

To evaluate the effectiveness of the S2CGAN model proposed in this article for highly imbalanced data, the study primarily focuses on the macro average index. The macro average treats each category equally, making it a reliable measure of performance for attacks with a small number of categories. The formulas for calculating the macro average index are provided in (12). By giving equal weight to each category, the macro average provides a comprehensive assessment of the classifier's performance in handling imbalanced scenarios, particularly for rare-level categories.

C. Experiment Procedure

This study utilizes the Keras and PyTorch frameworks to construct and evaluate the models. The experimentation is conducted on the Google Colaboratory Pro platform with Nvidia A100 GPU. The parameter settings and processing procedures for each module are described below.

The initial step in the preprocessing phase involves applying MAin-Max normalization to the CICIDS2017 data set, which restricts all numerical features to a range between 0 and 1. This normalization technique mitigates the influence of unit inconsistencies during the neural network training process. Subsequently, the data set is randomly partitioned into a training set and a test set, with the training set representing 80% of the total data.

Following the preprocessing phase, the training set is utilized to train the difference information extraction SAN module. The SAN module consists of two encoders and two decoders with shared parameters. The hidden layer of the SAN is configured as $64\times32\times16\times32\times64$, with the input and output layers comprising 78 neurons. The encoding dimension is set to 16, and the activation function employed is LeakyReLU. Each linear layer, excluding the output layer, is accompanied by BatchNorm1d, which enhances the connectivity within each batch. The batch size is specified as 64 to optimize the training process.

Following the categorization of the training set based on IR_i, the scarce-level attacks are passed through a pretrained SAN to extract features that serve as the conditional information s for the CGAN. In the CGAN architecture, the generator takes s and Gaussian noise z as inputs. The hidden layer parameters of the generator are configured as $32\times64\times128$, and the activation function employed for each layer, excluding the output layer, is LeakyReLU. The output layer utilizes the Sigmoid activation function. Each linear layer is accompanied by BatchNorm1d to facilitate batch-level connectivity.

The discriminator in the CGAN architecture consists of a hidden layer configured as 64×8. The activation function employed for each layer, excluding the output layer, is LeakyReLU, and the output layer utilizes the Sigmoid activation function. Each linear layer is followed by LayerNorm, which assists in normalizing the activations within each layer. The batch size for this module is set to 16 to optimize the training process.

Fig. 4 illustrates the dynamic shifts of loss profiles during the training process of both the SCGAN and conventional CGAN, both of which are executed under the same regular parameters. In the context of the SCGAN model, the loss functions associated with the generator and discriminator manifest stabilization of near predefined thresholds subsequent to multiple iterations of adversarial training. In contrast, within the conventional CGAN framework, the generator's potential to assimilate novel insights is constrained by the inhibitory influences imposed by the adversarial discriminator.

During the generation of scarce-level samples using the trained generator, we ensure sample quality by incorporating only those samples with a discriminator output exceeding 0.45 into the extended augmented data set. For rare-level attacks,

TABLE IV
EXPERIMENTAL RESULTS OF MULTICLASSIFICATION. INCLUDE THE ORIGINAL CLASSIFIER AND FOUR CLASS IMBALANCE ALGORITHMS

Methods	**Baseline**			**CVAE+GAN**			**SMOTE**			**TACGAN**			**S2CGAN**		
Metric	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score
BENIGN	0.9949	**0.9916**	**0.9932**	**0.9979**	0.9877	0.9928	0.9971	0.9847	0.9908	0.9965	0.9898	0.9931	0.9955	0.9903	0.9929
DoS/DDoS	0.9894	**0.9994**	0.9944	**0.9904**	0.9984	0.9944	0.9878	0.9993	0.9935	**0.9904**	0.9990	**0.9947**	0.9896	0.9993	0.9944
PortScan	**0.9082**	0.9369	0.9223	0.8671	**0.9818**	0.9209	0.8722	0.9644	0.9160	0.8878	0.9607	**0.9228**	0.9016	0.9417	0.9212
Patator	**0.9856**	0.9884	**0.9870**	0.9682	0.9892	0.9785	0.9845	0.9895	**0.9870**	0.9824	0.9895	0.9860	0.9433	**0.9924**	0.9672
Web Attack	0.9784	0.9358	0.9566	0.8881	0.8922	0.8902	0.2350	0.9404	0.3760	0.9000	0.9083	0.9041	**0.9794**	**0.9794**	**0.9794**
Bot	**1.0000**	0.3715	0.5417	**1.0000**	0.3740	0.5444	0.5746	0.7354	0.6451	0.9928	0.3486	0.5160	0.6091	**0.7455**	**0.6705**
Infiltration	0.0000	0.0000	0.0000	0.2500	0.1429	0.1818	0.8333	**0.7143**	0.7692	0.0000	0.0000	0.0000	**1.0000**	**0.7143**	**0.8333**
Heartbleed	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	**1.0000**	**1.0000**	**1.0000**	0.0000	0.0000	0.0000	**1.0000**	**1.0000**	**1.0000**

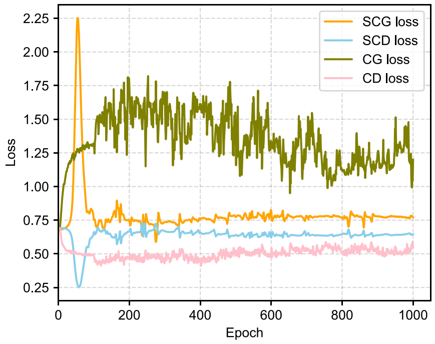

Fig. 4. Loss function of CGAN.

SKN is used to generate a sufficient number of samples (as Table III) to expand the augmented data set.

D. Comparative Experiments

This study entails a comparative analysis aimed at evaluating the performance of four algorithms for class imbalance. Our analysis encompasses the feature-based oversampling method SMOTE, as well as the data-based synthesis method conditional variational autoencoder (CVAE)-AN and the hybrid method tabular auxiliary classifier generative adversarial network (TACGAN). The algorithms under investigation are as follows:

Original: This refers to the raw data without any processing.

SMOTE: Tesfahun and Bhaskari [52] proposed SMOTE as a solution to address imbalanced data sets in machine learning. To generate a synthetic sample, SMOTE computes the difference between the feature vector of the considered instance and its nearest neighbor.

CVAE-AN: Sabeel et al. [53] introduced a novel adversarial incremental learning approach called CVAE-AN. This approach employs a CVAE to generate new samples by learning a distribution from the training data set. A Discriminator is then used to assess the quality of the generated samples based on how closely they resemble the original data set.

TACGAN: Ding et al. [54] introduced a TACGANs model to address the issue of imbalanced intrusion detection systems. The proposed approach combines undersampling and oversampling techniques to tackle class imbalance. To be more precise, the majority class is undersampled using the KNN algorithm, while GAN is utilized to oversample a limited number of minority class samples.

S2CGAN: The method proposed in this article.

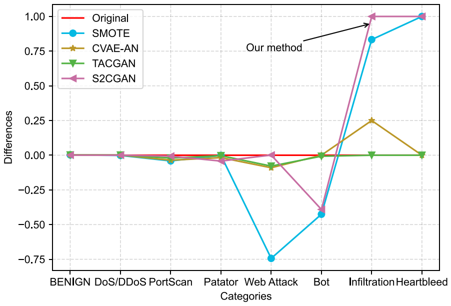

Fig. 5. Difference in Precision between each method and the original data set.

E. Discussion of Experimental Results

To evaluate the effectiveness of S2CGAN for imbalanced network intrusion detection, this article employs a basic DNN as the intrusion detection classifier. The classifier consists of four hidden layers with 128 × 64 × 32 × 16, and the activation function for each layer is Relu.

Additionally, the output layer of the classifier is set to the number of attack categories, and the activation function is Softmax. The loss function used in the classifier is categorical cross-entropy, and the optimizer is Adam. The batch size is set to 128, and the maximum number of epochs is 100.

As previously discussed, intrusion detection systems often encounter highly imbalanced network traffic, where accurate identification of each type of attack is crucial for effective preventive measures and enhanced network security. The detailed findings of the study are presented in Table IV, which provides a comprehensive overview of the results obtained from the evaluation of the proposed S2CGAN algorithm.

To facilitate a clear and comprehensive comparison of the performance of the four algorithms, we establish the IDS trained using the original data set as the baseline. By analyzing and comparing the differences between the baseline and the four methods, we can gain insights into the effectiveness of each algorithm in improving the detection performance of the IDS. The details of our findings are presented in Figs. 5–7, which depict the precision, recall, and F1-score, respectively. These figures provide a visual representation of the performance improvements achieved by each method across different attack categories, allowing for a comprehensive evaluation and comparison of their effectiveness in addressing the challenges posed by imbalanced network intrusion detection.

The performance analysis for the ample-level categories, as shown in the front sections of Figs. 5 and 6, reveals that there is minimal variation in performance among the four methods

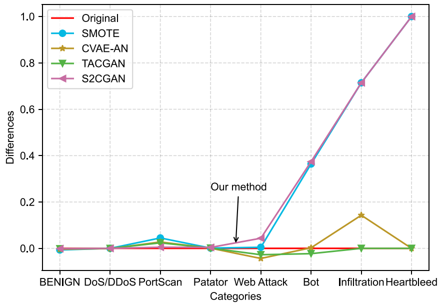

Fig. 6. Difference in Recall between each method and the original data set.

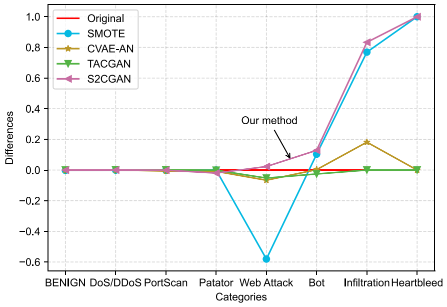

Fig. 7. Difference in F1-score between each method and the original method.

employed in this study. Our method takes into consideration that ample-level samples exhibit a relatively comprehensive distribution following category classification. Therefore, no additional processing is conducted on samples of this level to minimize computational costs. Since the other three methods do not involve a separate class classification step, they process both ample-level samples and other samples in the same manner. Despite the additional computational cost incurred by SMOTE, CVAE-AN, and TACGAN, the expected performance improvements have not been observed at this level.

Regarding the scarce-level categories, the middle sections of Figs. 5 and 6 demonstrate that both our proposed method and SMOTE exhibit a slight decrease in Precision for the *Patator* attack. Upon analyzing the scatter plot in Fig. 1, it is evident that the distribution of the *Patator* attack is highly concentrated. While our method successfully simulates the original distribution, it generates new points that lie outside the boundaries. However, it is reassuring that our method maintains a high Recall rate, ensuring the effective detection of the *Patator* attack.

In the case of other scarce-level attacks, the SCGAN algorithm proposed in this study leverages the original data distribution to simulate the possible distribution of these scarce-level categories. This can be observed in the intermediate sections of Figs. 5 and 6, where both Precision and Recall show improvements for the *Web Attack* and *Bot* categories. This indicates that our algorithm effectively captures and simulates the genuine data distribution of these two attack types, resulting in an enhanced detection rate for scarce-level attacks.

When considering rare-level attacks, the latter sections of Figs. 5 and 6 clearly show that both the baseline and TACGAN struggle to detect such attacks effectively. This is because the baseline and TACGAN rely solely on deep neural networks, which face difficulties in accurately representing the distribution of rare-level attacks due to the limited number of training samples available. On the other hand, CVAE-AN demonstrates a slight performance improvement as the variational encoder generates a larger number of slightly varied rare-level attacks. However, our method and SMOTE overcome the limitations of the data space by sampling from the eigenspace of rare-level attacks, taking into account the similarity of features among neighboring points in the eigenspace. The results clearly demonstrate that our method and SMOTE achieve outstanding performance in detecting rare-level attacks. Nonetheless, SMOTE might encounter challenges related to redundancy in real-world applications, leading to escalated computational load and diminished model generalization ability. In contrast, our enhanced lightweight SKN method circumvents the issue of excessive redundancy in SMOTE samples, rendering it better suited for extremely unbalanced IoT scenarios.

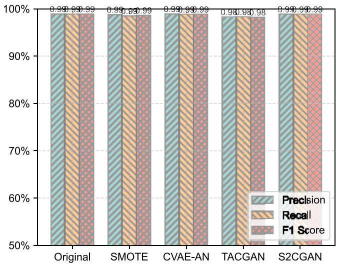

Fig. 8. Weighted average: Each class is given different weights according to the frequency of each class.

Fig. 1 provides a visual representation of the difference in comprehensive F1-score between the four methods and the baseline for each category. The figure clearly demonstrates the significant performance advantage of our method over other approaches for all minority categories. Our method consistently achieves a higher F1-score, indicating its effectiveness in accurately detecting and classifying attacks across various minority categories.

When assessing the overall performance of the entire data set, the weighted average is calculated by considering the frequency of occurrence of each category. This provides a comprehensive evaluation of the performance in unbalanced scenarios. Fig. 8 visually presents the comparison between our proposed framework and other methods across various categories. The figure indicates that our method performs similarly to other approaches regarding the weighted average, effectively ensuring a high detection rate for the overall data set.

In IoT scenarios with less frequent attacks, using a single weighted average metric may not adequately capture the performance of minority categories. To overcome this limitation, we utilize the macro-average metric, which specifically evaluates the detection performance of classifiers in unbalanced scenarios. The macro average index assigns equal weight to each category, enabling a more accurate assessment of classifier performance in unbalanced settings. As depicted in Fig. 9, the macro-average index confirms that our framework successfully addresses the challenge of extreme class imbalance in intrusion detection systems. Our method

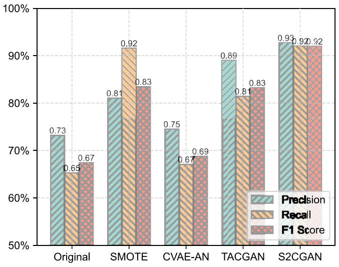

Fig. 9. Macro average: Each class is given the same weight.

demonstrates favorable outcomes even in highly imbalanced scenarios.

When comparing our method with the original classifier, significant improvements are observed in Precision, Recall, and F1-score. Specifically, our method demonstrates a remarkable increase of 22.05% in Precision, a substantial improvement of 43.81% in Recall, and a notable enhancement of 35.47% in F1-score.

These results emphasize the effectiveness and superiority of our method in detecting attacks in highly imbalanced IoT networks. Our approach demonstrates exceptional performance in accurately identifying and classifying attacks, even in scenarios where the class distribution is heavily skewed. This highlights the practical relevance and robustness of our method in improving the security and reliability of IoT networks.

VII. Conclusion and Future Work

Intrusion detection is a critical technology that is essential for ensuring IoT network security. However, the problem of extreme class imbalance can severely compromise the performance of intrusion detection systems. To address this issue, we propose a lightweight framework for intrusion detection called S2CGAN-IDS. Our framework includes a data processing module that categorizes network traffic into three levels: 1) ample; 2) scarce; and 3) rare, based on the degree of imbalance. We then employ an efficient SCGAN model to generate new scarce-level attacks and use simple SKN to oversample rare-level attacks. Finally, we train a simple DNN classifier with the augmented data set. Our experimental findings demonstrate the superiority of our method over other approaches for addressing the class imbalance, as evidenced by the macro average metric. Notably, our framework exhibits enhanced detection rates for nearly all minority classes while maintaining a high detection rate for the majority class.

The proposed algorithm demonstrates notable achievements in addressing specific data distributions; however, its performance may be limited when applied to diverse data types. To enhance its practicality, future work includes comprehensive testing on various data sets, a sensitivity analysis for optimal parameter configurations, and thorough validation to assess its generalization capabilities, aiming to improve its effectiveness in real-world scenarios and contribute to the advancement of the field.

In the future, we plan to explore additional classification models to further validate the effectiveness of our proposed scheme. Additionally, we acknowledge that this article is a case study in division category levels, and we aim to apply our S2CGAN-IDS to more general scenes in future research. By doing so, we hope to improve the generality and scalability of our proposed method so that it remains effective in more complex network environments.

References

[1] H.-J. Liao, C.-H. R. Lin, Y.-C. Lin, and K.-Y. Tung, "Intrusion detection system: A comprehensive review," *J. Netw. Comput. Appl.*, vol. 36, no. 1, pp. 16–24, Jan. 2013.

[2] A.-R. Mohamed, G. E. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 1, pp. 14–22, Jan. 2012.

[3] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1440–1448.

[4] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.

[5] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[6] E. Aminanto and K. Kim, "Deep learning in intrusion detection system: An overview," in *Proc. Int. Res. Conf. Eng. Technol. (IRCET)*, 2016, pp. 1–12.

[7] H. Liu and B. Lang, "Machine learning and deep learning methods for intrusion detection systems: A survey," *Appl. Sci.*, vol. 9, no. 20, p. 4396, Oct. 2019.

[8] Z. Ahmad, A. Shahid Khan, C. Wai Shiang, J. Abdullah, and F. Ahmad, "Network intrusion detection system: A systematic study of machine learning and deep learning approaches," *Trans. Emerg. Telecommun. Technol.*, vol. 32, no. 1, p. e4150, Jan. 2021.

[9] C. Liu, R. Antypenko, I. Sushko, and O. Zakharchenko, "Intrusion detection system after data augmentation schemes based on the VAE and CVAE," *IEEE Trans. Rel.*, vol. 71, no. 2, pp. 1000–1010, Jun. 2022.

[10] H. Zhang, X. Yu, P. Ren, C. Luo, and G. Min, "Deep adversarial learning in intrusion detection: A data augmentation enhanced framework," 2019, *arXiv:1901.07949*.

[11] S. H. Khan, M. Hayat, M. Bennamoun, F. A. Sohel, and R. Togneri, "Cost-sensitive learning of deep feature representations from imbalanced data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 8, pp. 3573–3587, Aug. 2018.

[12] C. Zhang, K. C. Tan, H. Li, and G. S. Hong, "A cost-sensitive deep belief network for imbalanced classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 1, pp. 109–122, Jan. 2019.

[13] S. Dhote, C. Vichoray, R. Pais, S. Baskar, and P. Mohamed Shakeel, "Hybrid geometric sampling and adaboost based deep learning approach for data imbalance in E-commerce," *Electron. Commer. Res.*, vol. 20, pp. 259–274, Jun. 2020.

[14] S. Aljawarneh, M. Aldwairi, and M. B. Yassein, "Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model," *J. Comput. Sci.*, vol. 25, pp. 152–160, Mar. 2018.

[15] P. Wu, H. Guo, and R. Buckland, "A transfer learning approach for network intrusion detection," in *Proc. IEEE 4th Int. Conf. Big Data Anal. (ICBDA)*, 2019, pp. 281–285.

[16] A. Singla, E. Bertino, and D. Verma, "Overcoming the lack of labeled data: Training intrusion detection models using transfer learning," in *Proc. IEEE Int. Conf. Smart Comput. (SMARTCOMP)*, 2019, pp. 69–74.

[17] J. Zhang, M. Zheng, J. Nan, H. Hu, and N. Yu, "A novel evaluation metric for deep learning-based side channel analysis and its extended application to imbalanced data," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2020, no. 3, pp. 73–96, 2020.

[18] P. Wei, Y. Li, Z. Zhang, T. Hu, Z. Li, and D. Liu, "An optimization method for intrusion detection classification model based on deep belief network," *IEEE Access*, vol. 7, pp. 87593–87605, 2019.

[19] A. A. Süzen, "Developing a multi-level intrusion detection system using hybrid-DBN," *J. Ambient Intell. Humaniz. Comput.*, vol. 12, no. 2, pp. 1913–1923, 2021.

[20] J. Kim, J. Kim, H. Kim, M. Shim, and E. Choi, "CNN-based network intrusion detection against denial-of-service attacks," *Electronics*, vol. 9, no. 6, p. 916, 2020.

[21] R. V. Mendonça et al., "Intrusion detection system based on fast hierarchical deep convolutional neural network," *IEEE Access*, vol. 9, pp. 61024–61034, 2021.

[22] M. T. Nguyen and K. Kim, "Genetic convolutional neural network for intrusion detection systems," *Future Gener. Comput. Syst.*, vol. 113, pp. 418–427, Dec. 2020.

[23] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," *IEEE Access*, vol. 5, pp. 21954–21961, 2017.

[24] M. Almiani, A. AbuGhazleh, A. Al-Rahayfeh, S. Atiewi, and A. Razaque, "Deep recurrent neural network for IoT intrusion detection system," *Simulat. Model. Pract. Theory*, vol. 101, May 2020, Art. no. 102031.

[25] S. Wang, W. Liu, J. Wu, L. Cao, Q. Meng, and P. J. Kennedy, "Training deep neural networks on imbalanced data sets," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, 2016, pp. 4368–4374.

[26] P. Bedi, N. Gupta, and V. Jindal, "Siam-IDS: Handling class imbalance problem in intrusion detection systems using siamese neural network," *Procedia Comput. Sci.*, vol. 171, pp. 780–789, 2020.

[27] P. Bedi, N. Gupta, and V. Jindal, "I-SiamIDS: An improved Siam-IDS for handling class imbalance in network-based intrusion detection systems," *Appl. Intell.*, vol. 51, pp. 1133–1151, Feb. 2021.

[28] N. Gupta, V. Jindal, and P. Bedi, "LIO-IDS: Handling class imbalance using LSTM and improved one-vs-one technique in intrusion detection system," *Comput. Netw.*, vol. 192, Jun. 2021, Art. no. 108076.

[29] N. Gupta, V. Jindal, and P. Bedi, "CSE-IDS: Using cost-sensitive deep learning and ensemble algorithms to handle class imbalance in network-based intrusion detection systems," *Comput. Secur.*, vol. 112, Jan. 2022, Art. no. 102499.

[30] S. Al and M. Dener, "STL-HDL: A new hybrid network intrusion detection system for imbalanced dataset on big data environment," *Comput. Secur.*, vol. 110, Nov. 2021, Art. no. 102435.

[31] K. M. Hasib, N. A. Towhid, and M. R. Islam, "HSDLM: A hybrid sampling with deep learning method for imbalanced data classification," *Int. J. Cloud Appl. Comput. (IJCAC)*, vol. 11, no. 4, pp. 1–13, 2021.

[32] J. Lee and K. Park, "GAN-based imbalanced data intrusion detection system," *Pers. Ubiquitous Comput.*, vol. 25, pp. 121–128, Feb. 2021.

[33] A. Ali-Gombe and E. Elyan, "MFC-GAN: Class-imbalanced dataset classification using multiple fake class generative adversarial network," *Neurocomputing*, vol. 361, pp. 212–221, Oct. 2019.

[34] F. Zhou, S. Yang, H. Fujita, D. Chen, and C. Wen, "Deep learning fault diagnosis method based on global optimization GAN for unbalanced data," *Knowl. Based Syst.*, vol. 187, Jan. 2020, Art. no. 104837.

[35] S. Huang and K. Lei, "IGAN-IDS: An imbalanced generative adversarial network towards intrusion detection system in ad-hoc networks," *Ad Hoc Netw.*, vol. 105, Aug. 2020, Art. no. 102177.

[36] J. Cui, L. Zong, J. Xie, and M. Tang, "A novel multi-module integrated intrusion detection system for high-dimensional imbalanced data," *Appl. Intell.*, vol. 53, no. 1, pp. 272–288, Jan. 2023.

[37] T.-T.-H. Le, Y. E. Oktian, and H. Kim, "XGBoost for imbalanced multiclass classification-based industrial Internet of Things intrusion detection systems," *Sustainability*, vol. 14, no. 14, p. 8707, 2022.

[38] J. L. Leevy, T. M. Khoshgoftaar, and J. M. Peterson, "Mitigating class imbalance for IoT network intrusion detection: A survey," in *Proc. IEEE 7th Int. Conf. Big Data Comput. Service Appl. (BigDataService)*, 2021, pp. 143–148.

[39] D. Stiawan, M. Y. B. Idris, S. Defit, Y. S. Triana, R. Budiarto, "Improvement of attack detection performance on the Internet of Things with PSO-search and random forest," *J. Comput. Sci.*, vol. 64, 2022, Art. no. 101833.

[40] I. Goodfellow et al., "Generative adversarial networks," *Commun. ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[41] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, *arXiv:1411.1784*.

[42] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," 2018, *arXiv:1802.05957*.

[43] S. Adiga, M. A. Attia, W.-T. Chang, and R. Tandon, "On the tradeoff between mode collapse and sample quality in generative adversarial networks," in *Proc. IEEE Glob. Conf. Signal Inf. Process. (GlobalSIP)*, 2018, pp. 1184–1188.

[44] Z. Zhang, M. Li, and J. Yu, "On the convergence and mode collapse of GAN," in *Proc. SIGGRAPH Asia Techn. Briefs*, 2018, pp. 1–4.

[45] L. Mescheder, A. Geiger, and S. Nowozin, "Which training methods for gans do actually converge?" in *Proc. 35th Int. Conf. Mach. Learn.*, 2018, pp. 3481–3490

[46] S. Zagoruyko and N. Komodakis, "Learning to compare image patches via convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2015, pp. 1–9.

[47] Y. Liu, X. Chen, H. Peng, and Z. Wang, "Multi-focus image fusion with a deep convolutional neural network," *Inf. Fusion*, vol. 36, pp. 191–207, Jul. 2017.

[48] X. Guo, R. Nie, J. Cao, D. Zhou, L. Mei, and K. He, "FuseGAN: Learning to fuse multi-focus image via conditional generative adversarial network," *IEEE Trans. Multimedia*, vol. 21, no. 8, pp. 1982–1996, Aug. 2019.

[49] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local Nash equilibrium," in *Proc. 31st Conf. Neural Inf. Process. Syst. (NIPS2017)*, 2018, pp. 1–38.

[50] A. Fernández, S. Garcia, F. Herrera, and N. V. Chawla, "SMOTE for learning from imbalanced data: Progress and challenges, marking the 15-year anniversary," *J. Artif. Intell. Res.*, vol. 61, pp. 863–905, Apr. 2018.

[51] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. ICISSp*, vol. 1, 2018, pp. 108–116.

[52] A. Tesfahun and D. L. Bhaskari, "Intrusion detection using random forests classifier with smote and feature reduction," in *Proc. Int. Conf. Cloud Ubiquitous Comput. Emerg. Technol.*, 2013, pp. 127–132.

[53] U. Sabeel, S. S. Heydari, K. Elgazzar, and K. El-Khatib, "CVAE-AN: Atypical attack flow detection using incremental adversarial learning," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, 2021, pp. 1–6.

[54] H. Ding, L. Chen, L. Dong, Z. Fu, and X. Cui, "Imbalanced data classification: A KNN and generative adversarial networks-based hybrid approach for intrusion detection," *Future Gener. Comput. Syst.*, vol. 131, pp. 240–254, Jun. 2022.