GCB-PPO2: A Hybrid Deep Reinforcement Learning Intrusion Detection System for Under-Represented Attack Categories in SDN

Chen Jue*, Tao Hongyu, Cui Meng, Peng Haidong, Qiu Xihe

***Abstract*—The centralized control plane inherent in SDN architecture creates critical security dependencies, where malicious exploitation of controller vulnerabilities could propagate systemic network failures. The Intrusion Detection System (IDS) effectively counters cyber threats, and Deep Reinforcement Learning (DRL) can enable the IDS to dynamically adapt to the constantly evolving attack patterns through autonomous environmental interaction and real-time policy optimization. However, the current DRL based IDS face the limitations of minority-class detection, generalization gap, and tuning complexity. This paper propose GCB-PPO2 (Generative Adversarial Networks CNNA-BiLSTM Proximal Policy Optimization 2), a hybrid DRL system synergistically combining Generative Adversarial Networks (GAN) and Proximal Policy Optimization 2 (PPO2). This framework integrates three innovations: (1) Unlike prior DRL models cannot detect low-frequency attacks correctly, we design a C-GAN (Conditional Generative Adversarial Networks) architecture to generate targeted under-represented attack samples for the SDN scenario intentionally, improving the recognition accuracy of the model for minority classes significantly. (2) Unlike existing DRL models that rely on single-modal networks and experiment on single datasets, we propose a PPO2-based DRL framework with a CNN-LSTM shared network to optimize dynamic policy adaptation while capturing spatio-temporal patterns, enhancing generalization ability to different environmental dynamics. (3) Unlike conventional DRL implementations that manually tune parameters or use random search, we embed a Bayesian optimization as a dedicated hyper-parameter auto-tuner, systematically resolving sensitivity bottlenecks and ensuring robustness in fluctuating environments. Experimental results demonstrate that GCB-PPO2 model achieves remarkable accuracy of 99.92% and 99.01% for binary and multiple classification scenarios, respectively, with F1-scores exceeding 85.91% for under-represented attack categories in InSDN dataset. Moreover, the model proves its strong generalization capability for maintaining over 98.65% accuracy on another dataset, while confirming its improved training efficiency for surpassing conventional hybrid deep learning models by more than 22.27% in training time.**

***Index Terms*—Software Define Network, Intrusion Detection, Deep Reinforcement Learning, Proximal Policy Optimization, Generative Adversarial Network, Convolutional Neural Network, Long Short Term Memory**

I. Introduction

Software Defined Network (SDN), characterized by its decoupled control plane and data plane architecture alongside logically centralized network management [1], addresses the limitations on scalability and flexibility of traditional networks, while catalyzing transformative advancements in network technology [2]. However, this architectural paradigm introduces heightened security risks inherent in its centralized control mechanisms [3]. In conventional distributed networks, security vulnerabilities typically exhibit localized impact, constrained by their decentralized topology structure [4]. Conversely, the logical centralization of SDN enables various risks such as single point of failure, protocol degradation attacks, and malicious logic implantation at the application layer. Each of these may lead to severe damage to the availability and integrity of the entire network. For instance, the degradation attacks against TLS (Transport Layer Security)/OpenFlow can force controllers to use insecure plain text channels, thereby eavesdropping or implanting malicious rules [5]. These systemic security implications necessitate that SDN protection constitutes a critical research priority [6].

Intrusion Detection Systems (IDS) have emerged as critical infrastructure for real-time network threat mitigation [7], particularly as conventional security measures like firewalls and heuristic analysis struggle to differentiate sophisticated attack patterns from legitimate traffic [8]. The integration of machine learning methodologies has revolutionized this domain through probabilistic modeling and statistical inference frameworks trained on extensive network behavioral datasets [4]. Representative implementations include the hybrid IDS architecture proposed by Al-Yaseen et al., which integrates Support Vector Machine (SVM) with extreme learning machine to optimize attack detection throughput [9]. Nevertheless, current machine learning-driven IDS implementations confront two fundamental limitations: intrinsic feature homogeneity across evolving attack variants and suboptimal feature representation capabilities in existing algorithms, which collectively induce incomplete pattern recognition and elevated false positive rates [10], [11].

Deep learning demonstrates superior efficacy over traditional machine learning in IDS, attributable to its automated feature abstraction capabilities and scalable data processing efficiency empowered by parallel computation architectures like Graphics Processing Units (GPU) [12], [13]. Our previous research develops the CNNA-BiLSTM (Convolutional Neural Network with Attention-Bidirectional Long Short Term Memory) framework, which is benchmarked on the InSDN dataset, a standardized SDN intrusion detection corpus. Experimental evaluations demonstrate accuracies of 99.86% and 96.66% for binary and multiple classifications, respectively, validating the architecture's discriminative capacity [14]. Nevertheless, deep learning implementations exhibit stringent dependency on voluminous high-quality training data with balanced class distributions [15]. Analysis of the InSDN dataset reveals extreme class imbalance, where high-volume attacks dominate,

while low-frequency attacks are severely under-represented. In this paper, DoS (Denial of Service), DDoS (Distributed Denial of Service), and Probe are classified as high-volume attacks, and U2R (User to Root), BFA (Brute Force Attack), Botnet, and Web are categorized as under-represented attacks. For instance, DoS attacks account for 24.90% of the total samples, whereas U2R attacks comprise merely 0.013% of the total samples, resulting in a three-order-of-magnitude disparity [16]. This skewed data distribution directly compromises the CNNA-BiLSTM model's detection efficacy for minority classes, particularly manifesting in complete failure to identify U2R intrusion patterns during validation trials.

Deep Reinforcement Learning (DRL) synergistically integrates the perceptual sophistication of deep neural network with the dynamic decision-making paradigms inherent to reinforcement learning [17], [18], thereby intrinsically mitigating inherent data dependency constraints [19], [20]. However, the related works on applying DRL to intrusion detection face the following critical challenges.

- Minority-class Detection: Under-represented attack categories exhibit sample scarcity, which are difficult for DRL models to classify correctly. For instance, the true-positive rate for U2R of the AESMOTE (Adaptive Enhanced Synthetic Minority Oversampling Technique) model is only 0.07% [21], while other three DRL models only conduct binary classification experiments [22]–[25].
- Generalization Gap: Existing studies have only proposed single-modal networks, including actor-critic [22], double DQN (Deep Q Learning) [21], or deep deterministic policy gradient [23], and have evaluated their performances on single datasets, including Gas Pipeline dataset [22], NSL-KDD (Network Security Laboratory Knowledge Discovery and Data Mining) [21], CIC-IDS2017 (Canadian Institute for Cybersecurity Intrusion Detection Systems 2017) [24], [25], or self-generated dataset [23], thereby failing to demonstrate sufficient generalization capability across diverse network environments or attack patterns.
- Tuning Complexity: Conventional DRL implementations rely on static hyper-parameters, causing instability in fluctuating environments. For instance, the hyper-parameters in papers [21], [23] are determined through manual trial and error combined with empirical selection, while papers [22], [24], [25] do not mention parameter setting.

To overcome the limitations of aforementioned related works, this study introduces GCB-PPO2 (Generative Adversarial Networks CNNA-BiLSTM Proximal Policy Optimization 2), a novel framework to improve the detection accuracy for minority class attacks, enhance the generalization ability, and reduce the tuning complexity for SDN intrusion detection. This framework dynamically assimilates network traffic patterns through three innovation pathways: (1) temporal-spatial feature co-learning via CNNA-BiLSTM hybrid architectures, (2) adaptive reward shaping mechanism through PPO2 [26], and (3) minority class augmentation by C-GAN (Conditional Generative Adversarial Networks) [27].

The main contributions and novelties of this paper are summarized as follows.

- We develop a customized C-GAN architecture to strategically augment under-represented attack categories for the InSDN dataset. Existing C-GAN based methods focus on traditional networks [28] or Unmanned Aerial Vehicles [29], [30], whereas our C-GAN is tailored for SDN-specific attack patterns and addresses the unique class imbalance in InSDN dataset. Moreover, as far as we know, we are the first paper to apply C-GAN to InSDN dataset without data pollution. This approach yields augmentation of under-represented attack instances with ratios ranging from $30.80\times$ to $288.75\times$ for the training set, while preserving the test set's original imbalance for fair evaluation, which is helpful for improving the recognition accuracy of the model for minority classes significantly.
- We propose a framework which integrates CNNA-BiLSTM architecture as a shared feature extractor within the PPO2 reinforcement learning paradigm. The combination of CNN and LSTM can model spatial correlation and temporal dynamics simultaneously, reducing the risk of overfitting caused by a single modal feature. The shared feature extractors of PPO2 framework enable reinforcement learning strategies to directly utilize optimized spatio-temporal features to enhance the generalization ability to different environmental dynamics.
- We pro-actively introduce Bayesian optimization as an autonomous hyper-parameter tuning module, building a probabilistic model of the objective function and then using this model to select the next most promising combination of hyper-parameters to find the optimal solution in as few iterations as possible. This strategy not only eliminates redundant feature processing layers to improve computational efficiency, but also enhances model robustness in dynamic network environments.
- We conduct comprehensive experiments to demonstrate the GCB-PPO2 model's capabilities through five key aspects: ablation studies confirming the individual contributions of C-GAN, PPO2, and shared network components; benchmark comparisons against traditional and state-of-the-art models showing consistent accuracy improvements; cross-domain evaluations on independent dataset verifying generalization capacity, supplemented by interpretability analysis through gradient-based feature visualization and robustness tests against adversarial perturbation scenarios.

Experimental results demonstrate that our proposed GCB-PPO2 model's triple advantages in SDN intrusion detection: high classification accuracy, strong generalization ability, and improved training efficiency.

- High Classification Accuracy: GCB-PPO2 model obtains the accuracies of 99.92% and 99.01% for binary and multiple classifications, respectively, and elevates the F1-score of under-represented attacks to over 85.91% for InSDN dataset.
- Strong Generalization Ability: GCB-PPO2 model maintains cross-domain effectiveness on NSL-KDD dataset,

achieving the accuracies of 99.66% and 98.65% for binary and multiple classifications, respectively.
- Improved Training Efficiency: GCB-PPO2 model shows 22.27% faster convergence speed than conventional hybrid deep learning models.

The rest of this paper is organized as follows. Section 2 concludes related works on IDS, including machine learning, deep learning, and DRL based methods. Our main contributions are described in the following two sections. Section 3 and 4 illustrate our proposed GCB-PPO2 model in detail, including phases of data augmentation and attack classification. In order to prove the effectiveness of GCB-PPO2 model, the simulation results of two phases are depicted in Section 5 and 6, respectively. Finally, concluding remarks are given in Section 7.

II. Related Works

The harm of cyber attacks to modern society should not be underestimated. Malicious intruders can undermine industrial control systems and trigger the shutdown of critical infrastructure. For instance, in September 2016, the Mirai botnet launched a large-scale DDoS attack on OVH company in France, with a peak traffic of 1 Tbps, completely crushing its network throughput capacity [31]. The IDS analyzes real-time traffic, captures abnormal behavior, and matches threat features, thereby issuing warnings in the initial phase of attacks. This enables emergency responders to secure the golden time window for mitigation [32]. In recent years, IDSs based on machine learning have developed rapidly and become increasingly mainstream due to its ability to handle complex data and adapt to new types of attacks [33]. Their technical frameworks can be refined into traditional machine learning, deep learning, and deep reinforcement learning models.

A. Related Works on Machine Learning Models

Due to its high accuracy, efficiency, and scalability, machine learning based models have gained an increasingly important role in modern network security protection [34]. Hadem et al. introduce a model combining SVM with selective logging for IP traceback, achieving a detection accuracy of 95.98% on the NSL-KDD dataset. Thereinto, selective logging is performed at the in-memory structure of controllers to save resources and enhance performance [35]. Yang et al. propose Griffin, an unsupervised machine learning model, raising the accuracy by at most 19% when compared with existing models. Griffin constructs feature extraction framework to capture sequential features, utilizes cluster analysis to reduce feature scales, and builds an ensemble auto-encoder to detect both known and zero-day intrusion attacks [6]. Lee et al. design a programmable switch based Intrusion Prevention System (IPS) on which machine learning algorithms run through utilizing switch CPU and pipeline to detect malware. Simulation results show that the proposed IPS increases the throughput and reduces the response time by $183X$ and 99.99% than SDN-based IPS, respectively [36]. El et al. make use of a boosting feature selection algorithm to select relevant Smart Grid (SG) based features, and put forward an ensemble learning model by combining several machine learning techniques together to detect SG based attacks, validating its efficiency on both NSL-KDD and UNSW-NB15 datasets [37]. Ahuja et al. import machine learning technique to classify benign traffics from DDoS attacks by combining SVM with random forest. Experiments depict that the proposed model classifies traffic with an accuracy of 98.8% along with a very low false alarm rate [38].

Although above mentioned works achieve certain effects in intrusion detection, there are still limitations when applying machine learning based models. Firstly, machine learning models often rely on artificially designed feature extractors which may not be able to fully cover all possible attack patterns, resulting in limited detection effectiveness [39]. Secondly, machine learning models may encounter challenges in computational complexity and storage requirements when faced with massive, high-dimensional network traffic data [40]. Thirdly, machine learning models may perform poorly at identifying unknown or novel types of attacks as they often learn based on characteristics of known attack patterns [41].

B. Related Works on Deep Learning Models

Unlike traditional machine learning, deep learning can overcome these disadvantages to some extent. Firstly, deep learning models can automatically learn feature representations from raw data, reducing the reliance on manual feature engineering [42]. Secondly, deep learning models can handle large datasets more efficiently with the help of distributed computing and GPU (Graphics Processing Unit) acceleration, speeding up the training process and improving detection performance [43]. Thirdly, deep learning models usually have a stronger generalization ability to identify new types of attacks due to their deep neural network structure and complex nonlinear mapping capability [44].

Zainudin et al. present a model for detecting DDoS attacks in Industrial Internet of Things (IIoT) networks, which apply the extreme gradient boosting for feature selection and a hybrid CNN-LSTM for attack classification. Performance results show that the model achieves the accuracy of 99.50% in CIC-DDoS2019 (Canadian Institute for Cybersecurity Distributed Denial of Service 2019) dataset [45]. Chanu et al. propose a voting-based hybrid feature selection technique which is cross-analyzed by three correlation methods, and then deploy a classifier combining multi-layer perceptron with genetic algorithm. Simulation results show that the model furnishes the accuracy and false positive rate by 98.8% and 0.6%, respectively [46]. Mhamdi et al. introduce a hybrid deep auto-encoder with a random forest classifier model to enhance intrusion detection performance. Moreover, authors design a three-layer framework for detecting and preventing attacks, consisting of an entropy-based detection method and aforementioned model running on the control plane as well as a proactive service monitoring program running on the application plane. Experiments depict that the framework achieves the anomaly detection rates by more than 98% [47]. Shu et al. utilize a collaborative IDS to detect abnormal network behaviors for the Vehicular Ad hoc Network (VANET), and conduct GAN

models on multiple SDN controllers to jointly train a global intrusion detection model [48]. As one of our previous works, we put forward a hybrid deep learning based IDS, which uses the genetic algorithm to extract positive features, and CNNA-BiLSTM model to classify different intrusion types, achieving accuracies of 99.86% and 96.66% on binary and multiple classifications, respectively [14].

It can be deduced that deep learning has a wider application and better performance in the field of intrusion detection when compared with traditional machine learning. However, deep learning still has shortcomings that need to overcome. Firstly, deep learning models often require large amounts of labeled data for training, which may be difficult to obtain or expensive to label, especially because network attack means are constantly updating [49]. Secondly, deep learning models, especially complex neural networks, often struggle to explain decision-making process, which makes it difficult to further optimize models [50]. Thirdly, deep learning models are prone to overfitting when faced with complex network environments and variable intrusion behaviors, leading to poor detection performance in practical applications [51].

C. Related Works on Deep Reinforcement Learning Models

Reinforcement learning, as an important branch of machine learning, has a stronger learning ability when compared with deep learning [52]. It is a multi-layer neural network similar to the structure of human brain, which can learn features from original data in different environments automatically, resulting in the achievement of optimal strategy [53]. Through combing deep learning with reinforcement learning, DRL has the abilities of strong perception and powerful decision-making simultaneously, which are beneficial to IDS [54]. On one hand, DRL models can learn optimal policies by interacting with environments, allowing it to adapt to changing network circumstances and intrusion behaviors [55]. On the other hand, DRL models can optimize response strategies by maximizing cumulative rewards, helping to make more accurate judgements in intrusion detection [24].

Wang et al. propose a DRL based model to detect abnormal flows for industrial control systems, which extracts features through neural networks and adjusts learning strategy according to reinforcement learning. Experimental results show that the model achieves the accuracy of 98.06% [22]. Ma et al. present an anomaly detection framework combining reinforcement learning with class-balance techniques, which uses the reinforcement learning to implement auto-learning and introduces an adapted Synthetic Minority Oversampling Technique (SMOTE) to address the class-imbalance problem. Simulation results on NSL-KDD dataset show that the accuracy and F1 score are greater than 82.0% and 82.4%, respectively [21]. Kim et al. build a secure network system by adopting the DRL to inspect network traffic and shuffling-based moving target defense technique to mitigate threats [23]. Yungaicela-Naula et al. provide a framework for detecting and mitigating slow-rate DDoS attacks automatically. The framework uses deep learning and reinforcement learning to detect and mitigate attacks, respectively, and includes a network function virtualization-assisted moving target defense mechanism to enhance the performance [24], [25].

TABLE I provides a structured comparison of state-of-the-art intrusion detection approaches, including their datasets, machine learning paradigms, and model architectures. By explicitly contrasting their advantage and limitations, the table underscores the methodological advancements of our work. The major differences in this research findings with already published works are summarized as follows.

- We propose a framework which integrates CNNA-BiLSTM architecture as a shared feature extractor within the PPO2 reinforcement learning paradigm to make full use of both techniques to achieve a high classification accuracy and an improved training efficiency. In contrast, existing DRL based IDSs only apply single-modal networks, including actor-critic [22], double DQN [21], and deep deterministic policy gradient [23], to detect or classify network attacks.
- We adopt the Bayesian optimization to implement automated hyper-parameter tuning to make the model more accessible and reliable. In contrast, existing DRL based IDSs employ random search for hyper-parameter optimization [24], [25] or rely on manual parameter tuning [21], [23].
- We conduct comprehensive experiments to demonstrate our model's performance on both the binary and multiple classification scenarios, as well as for both common and under-represented attack categories. In contrast, existing DRL based IDSs do not consider multiple classification [22]–[25] or cannot classify under-represented attacks efficiently [21].

III. Proposed Methodology of Data Augmentation

The overall workflow diagram of our proposed GCB-PPO2 is shown in Fig. 1. Each SDN controller runs a network monitoring service to capture data traffic from OpenFlow switches through TCPDump and Wireshark tools, including malicious and benign traffics. Prior to providing data to attack classification model, three steps are required to ensure optimal data quality. Firstly, data preprocessing is performed including data cleaning and standardization to improve data consistency. Secondly, genetic algorithm is applied to select potential features and reduce feature dimension, thus significantly reducing the model complexity. Readers can refer to our previous work in [14] for more details of these two steps. Thirdly, data augmentation based on C-GAN is conducted to expand the dataset by generating samples of under-represented attack categories intentionally, which can overcome the class imbalance problem, thus beneficial for further raising the accuracy of detecting minority class traffics. The core module of this system is the attack classification model by constructing a DRL environment based on PPO2 which takes CNNA-BiLSTM as the shared network, thus achieving high performance of identifying and classifying network attacks. The details of data augmentation and intrusion detection will be introduced in the following two sections, respectively.

Unbalanced data is a non-negligible factor reducing the classification performance of IDS. For instance, our previously

TABLE I: Comparison Table among Related Works

Literature	Dataset	Machine Learning Paradigm	Model Architecture	Advantage	Limitation
[35]	NSL-KDD	Machine Learning	SVM	Save memory by logging anomalous packets only	Lack of multiple classification
[6]	Dataset provided by Yisroel Mirsky	Machine Learning	Auto-Encoder	Update training model in an unsupervised manner dynamically	Lack of underrepresented attacks in dataset
[36]	IoT-23	Machine Learning	Neural Network	Implement high-throughput programmable switch-based prevention system	Lack of multiple classification
[37]	NSL-KDD, UNSW-NB15	Machine Learning	Ensemble Learning	Import feature selection to optimize training/test time	Lack of multiple classification
[38]	Self-generated dataset	Machine Learning	SVM, Random Forest	Create SDN dataset	Restriction to DDoS attacks
[45]	CIC-DDoS2019	Deep Learning	CNN, LSTM	Implement low-cost DDoS classification	Restriction to DDoS attacks
[46]	CIC-IDS2017, NSL-KDD, DARPA	Deep Learning	Multi-Layer Perceptron, Genetic Algorithm	Import voting-based feature selection with no redundancy	Restriction to DDoS attacks
[47]	CIC-IDS2017	Deep Learning	Auto-Encoder, Random Forest	Implement three-layer protection mechanism	Restriction to DDoS attacks
[48]	KDD99, NSL-KDD	Deep Learning	GAN	Train global model with multiple SDN controllers	Lack of multiple classification
[14]	InSDN	Deep Learning	CNN, LSTM	Implement accurate multi-classification detection	Long response time required
[22]	Gas Pipeline Dataset	DRL	Actor-Critic	Combine perception with decision-making	Lack of multiple classification
[21]	NSL-KDD	DRL	Double DQN	Improve class-imbalance problem	Lack of underrepresented attacks
[23]	Self-generated dataset	DRL	Deep Deterministic Policy Gradient	Implement proactive intrusion prevention mechanism	Lack of multiple classification
[24], [25]	CIC-IDS2017	DRL	LSTM	Import reinforcement learning to mitigate attacks and network function virtualization to assist moving target defense	Lack of multiple classification

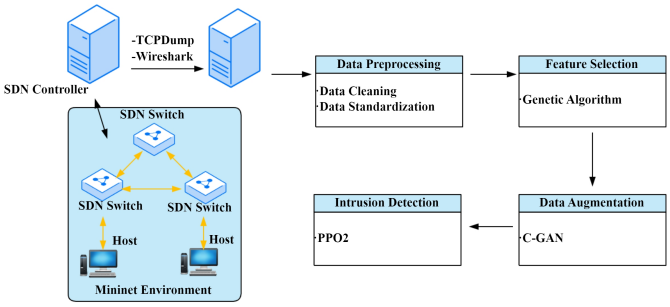


Fig. 1: Workflow Diagram of GCB-PPO2

proposed CNNA-BiLSTM model cannot detect U2R, Botnet, and Web attacks effectively, as inadequate number of samples result in inadequate training [14]. The common idea to counter this problem is to generate data for minority classes through data generation algorithms, so as to balance the dataset by means of expansion and reconstruction. Data augmentation algorithms usually include Variational Auto Encoders (VAE) and GAN. VAE generates samples by learning potential representations of data, often producing samples which are highly consistent with the distribution of original samples. However, network attacks aiming at SDN are complex and changeable, while attack samples generated by VAE often lack sufficient diversity, potentially leading to overfitting problem during the training process [56].

GAN, on the other hand, uses real data as input and generates synthetic data through a generator, which is often applied to scenarios where obtaining data is difficult, time-consuming, or expensive. Two key components constitute GAN: a generator and a discriminator. Generator is responsible for processing input data and generating more synthetic data, with the goal of simulating the patterns and characteristics of real data. While discriminator is a classifier whose job is to identify whether the input data is real or synthetic, and to output a probability representing the authenticity of samples. The probability close to 1 means samples are more likely to be real, while the probability close to 0 means samples are probably generated [57].

In traditional GAN, the generator generates samples only

based on random noise and has no control over the class or properties of generated samples. In contrast, C-GAN guides the generation process by adding conditional variables such as class labels, resulting in more controlled samples [58]. Therefore, C-GAN has stronger control and interpretability. On closer analysis, there are existing research works which allows C-GAN based attack dataset generation for various use cases. For instance, Zeng et al. introduce the Conditional Tabular GAN (CT-GAN) to generate synthetic samples to enhance the diversity and representativeness of Unmanned Aerial Vehicles (UAV) dataset [29], [30]. Wang et al. use CT-GAN to expand the small category traffic samples and balance the dataset to improve the malicious traffic identification rate for traditional networks and the CIC-IDS2017 dataset [28]. As far as we know, there is only one paper applying C-GAN to SDN intrusion detection [12]. However, authors of [12] fundamentally mix synthetic and real-world data during testing phase, creating artificial data distribution overlaps that systematically overestimate model performance, and violating the essential machine learning principle of strict train-test separation when dealing with generative models.

In contrast, our work presents three key contributions on applying C-GAN to SDN based IDS.

- We are the second research initiative exploring C-GAN applications in SDN environments for cybersecurity purposes, addressing the notable research gap in this emerging domain.
- We design a C-GAN framework to synthesize targeted under-represented attack samples intentionally for InSDN dataset, resolving extreme class imbalance and enabling models to learn discriminative features of minority class samples. Unlike prior work [12] that forcibly balance the sample size, which might introduce problems such as distribution distortion, noise interference and evaluation bias.
- We confine synthetic data strictly to the training phase, with original test data preserved, eliminating performance inflation from data leakage and ensuring credible evaluations, which is absent in prior work [12].

As a result, the generated samples are expected to improve the classification performance of minority class samples, as validated in Section VI.B.

InSDN dataset is uniquely designed for SDN environments, capturing the distinct architecture of SDN with its centralized control plane and separate data plane. It collects real network traffic and covers common attack types in SDN networks, reflecting the specific control and forwarding behaviors of SDN networks. On the other hand, the distribution of different attack types in the dataset is highly unbalanced, reflecting the attack distribution in the actual network. Moreover, InSDN dataset is widely used in the performance evaluation of IDS, and its design is in line with the actual operation and maintenance requirements.

Facing the InSDN dataset, the workflow diagram of C-GAN is shown in Fig. 2. Firstly, the dataset has gone through data preprocessing and feature selection steps, which are omitted in this figure. Then the dataset is divided into two parts: attack samples of under-represented categories and those of other

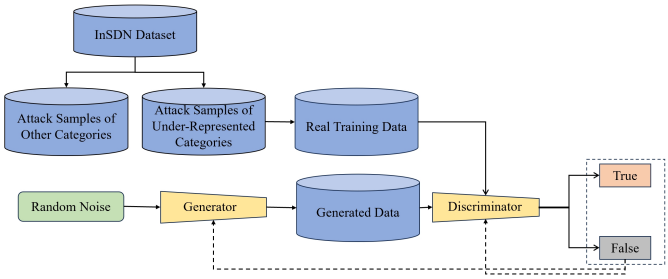


Fig. 2: Workflow Diagram of C-GAN

categories. Thirdly, C-GAN is constructed, whose generator receives not only random noise but also label information of specific class samples as conditional input, while the discriminator receives both real and generated data as well as their corresponding label information.

During training process, the goal of the generator is to generate samples of synthetic data similar to real data, and its loss function can be expresses as:

$$L_G = -E_{z,c}\left[\log D\left(G\left(z,c\right)\right)\right] \tag{1}$$

where z represents random noise, which is used as the input of the generator and can be understood as the seed or random factor for generating malicious traffics, c represents a tag, and $G(z,c)$ represents samples generated by the generator based on the random noise and label. On that basis, $D(G(z,c))$ outputs a probability to judge whether the sample $G(z,c)$ is real, and $E_{z,c}[\cdot]$ denotes the expectation of the joint distribution of random noise and label. As a result, the formula aims to maximize the probability that discriminator determines that the generated sample is real. Through training, the generator tries to generate samples more realistic, so that the discriminator cannot tell them apart from real ones.

On the other hand, the discriminator attempts to distinguish between real data and generated data for under-represented attack samples. As the generator learns through repeated iterations how to generate more realistic samples of under-represented attack categories, the discriminator continues to adjust to better identify the difference between real data and synthetic data. The loss function of discriminator can be expressed as:

$$L_D = -E_{x,c}\left[\log D\left(x,c\right)\right] - E_{z,c}\left[\log\left(1 - D\left(G\left(z,c\right),c\right)\right)\right], \tag{2}$$

where x represents a known real sample, including normal or attack sample, and $D(x,c)$ outputs a probability to judge whether the sample is real. $\log\left(1 - D\left(G\left(z,c\right),c\right)\right)$ is responsible for calculating the logarithm of the probability that the generated sample is judged to be false. As a result, the first and second expected value terms denote the discriminator's correct classification probabilities for real samples and generated samples, respectively, and the loss function is used to train the discriminator with the goal of maximizing their sum. Moreover, the dropout layer is added and placed after the fully connected layer in the discriminator as a means of regularization to prevent it from relying too heavily on certain specific features, improving the models's ability to generalize and avoiding overfitting problem.

Conditional labels implicitly divide the feature space to ensure that generated samples of different categories form separate clusters in the feature space. The generator learns the feature distribution of different attack categories through conditional labels $p_{\text{gen}}(x|c)$. Through adversarial training, the generator is forced to approximate the conditional distribution of real data $p_{\text{data}}(x|c)$. Mathematically, the degree of proximity between the generated distribution and the true distribution can be measured by KL (Kullback-Leibler) divergence:

$$\text{KL}\left(p_{\text{data}}(x|c) \parallel p_{\text{gen}}(x|c)\right) = \mathbb{E}_{x\sim p_{\text{data}}}\left[\log \frac{p_{\text{data}}(x|c)}{p_{\text{gen}}(x|c)}\right]. \quad (3)$$

When the adversarial training converges, the KL divergence approaches 0, indicating that the generated samples can represent the statistical characteristics of the real data.

In C-GAN, the generator and discriminator are optimized through adversarial training, with the goal of making the generated samples as close as possible to the real data distribution. According to the Nash Equilibrium theory in game theory, C-GAN will converge to a Nash equilibrium under ideal circumstances. At this equilibrium point, the generator can no longer "deceive" the discriminator by generating different samples, and the discriminator can no longer effectively distinguish between real data and generated data. Therefore, the distribution of the generated samples will cover that of the real data, achieving a "match" between the generated samples and the real data. The pseudo-code of our proposed C-GAN is illustrated in Algorithm 1.

Algorithm 1 Pseudo Code of C-GAN for Under-Represented Attack Categories of InSDN Dataset

Require:
primary InSDN dataset D_{real}, generator network G (see Table IV), discriminator network D (see Table V), learning rate α, batch size m, latent dimension $z_{\dim}$, epochs N, save interval k (e.g. every 100 epochs)
Ensure: augmented dataset D_{aug}
initialize G and D parameters;
for epoch $= 1$ to N **do**
  shuffle D_{real} to create random batches;
  for each mini-batch sampled from D_{real} **do**
    sample random noise;
    sample conditional labels;
    generate fake data using G;
    compute loss for real data;
    compute loss for fake data;
    update D by back-propagating combined real and fake data;
    sample new random noise and conditional labels;
    generate new fake data;
    compute loss for G;
    update G by back-propagating loss to improve generated data's realism;
  end for
  if epoch $\bmod k = 0$ **then**
    save checkpoint of G (and optionally D);
  end if
end for
choose desired number n_{aug} of synthetic minority samples;
sample random noise $\{z_j\}_{j=1}^{n_{\text{aug}}}$ and conditional labels $\{c_j\}_{j=1}^{n_{\text{aug}}}$;
generate fake data;
combine generated data with real data;
return D_{aug};

IV. Proposed Methodology of Attack Classification

As one of DRL algorithms, Proximal Policy Optimization (PPO) is proposed by Google DeepMind and OpenAI in 2018, which aims to improve convergence speed and sampling efficiency while maintaining stability. PPO is composed of four components, including advantage function, proximal policy optimization, multi-step sampling and importance sampling, which are illustrated as follows.

- Advantage Function: PPO adopts it to evaluate the degree of advantage of taking an action relative to the average situation, so as to guide the direction of policy update.
- Proximal Policy Optimization: PPO uses a method of cutting to limit the amplitude of each policy update.
- Multi-Step Sampling: PPO typically chooses it to estimate the value of an action, which helps to improve sampling efficiency.
- Importance Sampling: PPO often imports it to estimate the expected benefits of policy updates to make better use of historical data.

In a conclusion, PPO algorithm achieves stable training in DRL tasks by limiting the update amplitude, increases convergence speed by maximizing the advantage function at each time, and improves sampling efficiency by combining multi-step sampling with importance sampling [59]. As an improved version of PPO, PPO2 mainly optimizes the clipping mechanism, objective function, and implementation details, thus further improving the performance and stability of the algorithm. Based on the above considerations, this paper constructs a PPO2 based DRL environment for intrusion detection of SDN.

PPO2 is an actor-critic based algorithm, and the loss function of policy network can be expressed as:

$$A_{\_loss} = \hat{E}\left[\min\left(ratio \times \hat{\mathrm{A}}_t, clip\left(ratio, 1-\varepsilon, 1+\varepsilon\right)\hat{\mathrm{A}}_t\right)\right], \quad (4)$$

where $ratio$ and $\hat{A}_t$ represent the strategy ratio and advantage function, which will be introduced later. $clip\left(ratio, 1-\varepsilon, 1+\varepsilon\right)$ denotes the clipping operation, limiting the policy ratio to the range $[1-\varepsilon, 1+\varepsilon]$ to avoid instability caused by excessive policy update. Thereinto, ε represents a hyper-parameter controlling the magnitude of policy updates. In all, this loss function implements the policy update by selecting the minimum of two terms: (1) the untrimmed policy update item $ratio \times \hat{\mathrm{A}}_t$, which represents the advantage function adjusted by the current policy ratio; (2) the trimmed policy update item$clip\left(ratio, 1-\varepsilon, 1+\varepsilon\right)\hat{\mathrm{A}}_t$,

which controls the update amplitude by restricting the policy ratio within a bounded range. This design enhances the stability of policy updates.

In formula (4), $\hat{A}_t$ represents the advantage function, which is used to estimate the advantage at time step t, measuring how good or bad the current action is relative to other possible actions. It can be calculated through the difference value of action value function and state value function by Generalized Advantage Estimation (GAE):

$$\hat{A}_t = Q_\pi(s,a) - V_\pi(s). \tag{5}$$

This formula measures how good or bad it is to take a particular action a in state s relative to the average (given by $V_\pi(s)$) in that state. $ratio$ represents the strategy ratio, and can be calculated through:

$$ratio = \frac{p_\theta\left(a_t|s_t\right)}{p_{\theta old}\left(a_t|s_t\right)}, \tag{6}$$

which is the probability ratio of new and old strategies to choose action a in a given state s at time step t. Here, the numerator and denominator represent the new and old strategies to choose action a in a given state s, respectively.

As a result, the clipping mechanism of PPO2 ensures that the proportion of each policy update does not exceed a set range, thereby guaranteeing the monotonic improvement of each update. In PPO2, the objective function measures the effectiveness of the update by comparing the probability ratio between the old and new strategies. When the probability ratio exceeds the set threshold, the objective function will be trimmed to limit the update range, thereby avoiding excessive policy updates. According to the monotonic improvement theorem of PPO2, this clipping mechanism ensures the stability of policy updates and prevents performance deterioration caused by policy updates. Through this pruning mechanism, PPO2 can maintain the stable improvement of the strategy in multiple training steps and eventually converge to an optimal strategy.

On the other hand, the loss function of critic network can be calculated through the mean squared error:

$$C\_loss = mean\left(square\left(\hat{A}_t\right)\right). \tag{7}$$

Finally, the loss function for the whole PPO2 algorithm is:

$$PPO\_loss = A\_loss + 0.5 \times C\_loss, \tag{8}$$

which needs to be minimized through optimization algorithms to update parameters of policy network.

In this paper, the DRL process of PPO2 algorithm is divided into two stages: interaction stage and learning stage. The main purpose of the interaction stage is to collect training data. In this stage, agents interact with the environment, collect information about the current state, and make decisions based on it and existing policies. At the same time, the environment provides agents with rewards and information about the next state. This process is repeated several times to gather enough data for the learning stage. The workflow diagram of PPO2's interaction stage is shown in Fig. 3. The main purpose of the learning stage is to use the data collected during the interaction stage to update policies to improve agents' performance, that is, agents learn and update the policy network based on sample data from experience pool containing previous states, current states, policies and rewards.

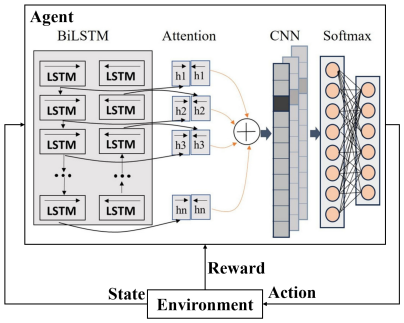


Fig. 3: Workflow Diagram of PPO2's Interaction Stage

TABLE II: Action Space of Binary Classification

Action	Definition
No Alert	0
Alert (Attack)	1

The "state" of DRL refers to the environment information observed by the agent at each step, that is, the input data provided by the environment to the agent. In this paper, the "state" is extracted from every row of data (except the label column) in the dataset. As mentioned in our previous work [14], after data preprocessing and feature selection, 12 features are reserved for CNNA-BiLSTM model, including "Bwd Pkt Len Min", "Bwd Pkt Len Std", "Flow Byts/s", and so on. For instance, "Bwd Pkt Len Min" means reverse minimum packet length, which indicates the minimum length of a single packet in the reverse traffic direction. It is obtained by aggregating the minimum length of all reverse packets in the session flow, and is used to capture extreme events in response traffic, such as control commands or tiny packets. If very small packets frequently appear in reverse traffic, it may indicate covert communication behaviors, such as network scanning and heartbeat detection, which are commonly seen on the botnet control channel. As a result, each state vector is an eigenvector of data with 12 features after normalization, representing relevant features of network traffic once, and the agent's goal is to make the right decision based on these features.

The definitions of "action" differ in binary and multiple classification scenarios. In binary classification, the defined action space is shown in TABLE II. In multiple classification, the defined action space is shown in TABLE III. It can be observed that an action is the classification result of an intrusion detection.

The PPO2 algorithm used in this paper is based on actor-critic algorithm, in which intrusion detection agents train policy network as actors and predict the policies that should be executed at present, with the goal of maximizing the future cumulative rewards. In order to update neural networks in time, it is necessary to fuse policy network with critic network, so that our previously proposed CNNA-BiLSTM is shared between two networks, and the weights of two networks can be shared and updated at the same time. Moreover, PPO2 uses

TABLE III: Action Space of Multiple Classification

Action	Definition
No Alert	0
Alert (DoS Attack)	1
Alert (DDoS Attack)	2
Alert (Probe Attack)	3
Alert (U2R Attack)	4
Alert (BFA Attack)	5
Alert (Botnet Attack)	6
Alert (Web Attack)	7

a fixed-length trajectory segment, representing a sequence of states and actions. In each iteration, N parallel agents collect T steps of data, and the loss function is constructed on NT steps and optimized by a small-lot gradient descent or Adam optimizer. The pseudo-code of PPO2 algorithm is described in Algorithm 2.

Algorithm 2 Pseudo Code of GCB-PPO2 Algorithm

Require:
expanded InSDN dataset, policy network, critic network
for each iteration **do**
  initialize intrusion detection environment and collect initial state s_0;
  initialize empty experience pool;
  for each step in episode **do**
    output action $a_t \sim \pi_\theta(a_t|s_t)$;
    execute a_t and observe s_{t+1}, r_t;
    store (s_t, a_t, r_t, s_{t+1}) into experience pool;
    update state s_t to s_{t+1};
  end for
  if sufficient data in experience pool **then**
    sample a batch of transitions from experience pool;
    compute action value and state value functions $Q_\pi(s,a)$ and $V_\pi(s)$ for each transition;
    compute advantage function $\hat{A}_t$ based on $Q_\pi(s,a)$ and $V_\pi(s)$ using GAE;
    compute clipped surrogate loss A_{loss};
    update policy network p_θ to minimize A_{loss};
    compute loss function of critic network C_{loss};
    update critic network to minimize C_{loss};
  end if
  set $p_{\theta old}$ to be p_θ;
end for

Even though our previously proposed CNNA-BiLSTM achieves good classification results in intrusion detection and classification of SDN, it still needs to be improved especially when facing under-represented attack categories. In this paper, we innovatively combine DRL with deep neural network to make advantage of both techniques, i.e., constructing a PPO2 environment which uses CNNA-BiLSTM as the shared network. Unlike supervised learning methods like CNNA-BiLSTM, PPO2 constantly updates its strategy through interaction with environment to learn how to make optimal decisions in a changing environment without the need to define fixed rules or structures in advance. As a result, PPO2 is appropriate to unknown or fast-changing environments through its online learning and continuous improvement. For instance, once PPO2 has been trained well in InSDN dataset, it can be ported to other intrusion detection datasets adaptively.

Moreover, CNNA-BiLSTM is very efficient at extracting spatial features (by CNN) and temporal information (by BiLSTM) from high-dimensional data. After integrating CNNA-BiLSTM into PPO2, agents can better understand spatial and temporal characteristics of input data when dealing with complex and time-dependent environments, so as to make more intelligent decisions and enhance the overall performance. Moreover, joint feature extraction by CNNA-BiLSTM can reduce the data dimension and make the input information processed by PPO2 more compact, thus reducing the computational burden. As CNNA-BiLSTM completes deep feature extraction in advance, so that PPO2 only needs to learn the strategy and does not need additional complex feature learning layer, thus reducing the calculation amount of the strategy network. As a result, this shared network can promise a high classification accuracy as well as an improved training efficiency.

V. Performance Evaluations of Data Augmentation

A. Experimental Setup of Whole System

In terms of hardware environment, all the deep learning and DRL models used in this paper are trained and tested under the support of GPU on the CentOS operating system, with the CPU being Intel(R) Xeon(R) Gold 5118 CPU @ 2.30 GH, and GPU being Nvidia GeForce RTX 2080Ti 11GB. In terms of software environment, we choose Python language and use Conda to manage Python environments and dependency packages. The implementation is primarily based on PyTorch deep learning framework with CUDA acceleration for GPU computing. For reinforcement learning components, we utilize Stable-Baseline library, which provides a range of easy-to-use and efficient algorithm implementations designed to simplify the process of setting, training, and evaluating. Moreover, the machine learning pipeline incorporates scikit-learn for data preprocessing and performance evaluation.

The structures of the generator and discriminator of C-GAN are summarized in TABLE IV and TABLE V, respectively. In terms of the generator architecture, the network begins with an input layer accepting 1000-dimensional vectors ($batchsize = 64$). The core structure contains four consecutive fully connected layers with ReLU activation. The first dense layer expands the feature dimension to 128, followed by progressive expansion to 256, 512, and finally the last layer outputs 19-dimensional features. Notably, the layer expansions do not strictly follow the "doubling" pattern described, with the final layer actually performing dimension reduction.

In terms of the discriminator design, the network begins with a 19-dimensional input layer ($batchsize = 64$) that receives both real and generated samples. The core architecture comprises three sequential fully connected layers with ReLU activation, where the first two layers are accompanied by dropout regularization. The feature dimension evolves through 512, 256, and 128 neurons in the hidden layers. The last layer

TABLE IV: Structure of Generator

Layer	Type	Output Size	Activation Function	Parameter
Input_1	Input Layer	(64,1000)	-	0
Dense	Dense	Multiple	ReLU	128128
Dense_1	Dense	Multiple	ReLU	33024
Dense_2	Dense	Multiple	ReLU	131584
Dense_3	Dense	Multiple	ReLU	9747

TABLE V: Structure of Discriminator

Layer	Type	Output Size	Activation Function	Parameter
Input_2	Input Layer	(64,19)	-	0
Dense_4	Dense	Multiple	ReLU	10240
Dropout	Dropout	Multiple	-	0
Dense_5	Dense	Multiple	ReLU	131328
Dropout_1	Dropout	Multiple	-	0
Dense_6	Dense	Multiple	ReLU	32896
Dense_7	Dense	Multiple	Sigmoid	129

of the discriminator is still the fully connected layer with Sigmoid as the activation function used for identifying the authenticity of samples.

Considering that the performance of our proposed GCB-PPO2 relies heavily on specific hyper-parameters, we adopt Bayesian optimization, i.e., one of the automated hyper-parameter tuning methods, to make the model more accessible and reliable. Bayesian optimization works by building a probabilistic model of the objective function (usually a Gaussian process) and then using this model to select the next most promising combination of hyper-parameters to find the optimal solution in as few iterations as possible. Compared to grid search or random search, it is more efficient, especially in high-dimensional spaces and complex dependencies. Moreover, Bayesian optimization is also expected to enhance the robustness of GCB-PPO2 model by systematically selecting hyper-parameters resistant to perturbations for the following two reasons. On one hand, the Gaussian process surrogate identifies wide, high-performance regions, where small parameter perturbations cause minimal performance drop. On the other hand, the acquisition function favors parameters with low validation accuracy variance, ensuring consistent performance under environmental noise.

We choose Optuna as the Bayesian optimization library, as it is more modern and is easier to integrate with existing code. In order to perform hyper-parameter optimization, the training process needs to be wrapped into an objective function that takes in hyper-parameters and returns evaluation metrics. The Bayesian optimizer then continuously calls this objective function, adjusting the parameters to maximize metrics. Fig. 4 shows the model's validation accuracy at each successive Bayesian optimization trail, as well as the validation accuracy at different values of each hyper-parameter. It can be observed from Fig. 4j that after a few initial exploratory trials where accuracy fluctuates apparently, the optimizer quickly homes in on high-performing regions: most subsequent trials achieve accuracies above 0.95. Although there are occasional small dips, the overall trend is convergent, with the best trails reaching 0.9899. Other figures plot each Bayesian optimization trial on the x-axis against one hyper-parameter on the y-axis, with point color encoding the resulting validation accuracy. For instance, when $batch\_size$ equals to 64, the highest validation accuracy of the model is achieved. Finally, the derived hyper-parameters are summarized in TABLE VI.

TABLE VI: Hyper-Parameter Setting of GCB-PPO2 Framework

Name	Meaning	Value
batch_size	Minibatch size	64
clip_range	Clipping parameter ϵ	0.2477
ent_coef	Entropy regularization coefficient	0.0268
gae_lamda	GAE parameter λ	0.9690
gamma	Discount factor	0.9135
learning_rate	Learning rate	0.0002
n_epochs	Optimization epochs per update	26
n_steps	Steps per environment per rollout	256
targer_kl	Measure difference between probability distributions	0

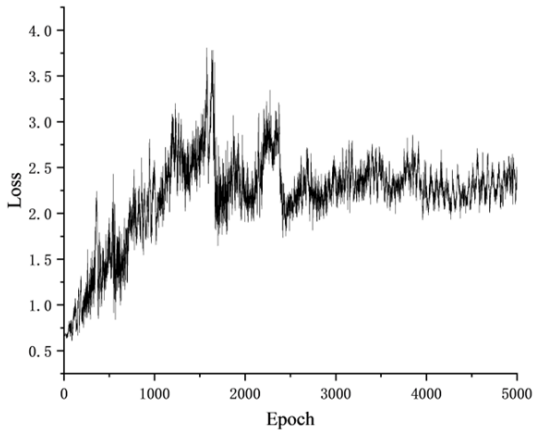


Fig. 5: Loss Curve of Generator

B. Experimental Results of Data Augmentation

As mentioned before, this paper adopts C-GAN to generate under-represented attack class samples to expand InSDN dataset after data preprocessing and feature selection. The loss curve of generator for the training process is shown in Fig. 5. It can be observed that at the beginning of training (around the first 1500 epochs), the loss value rises rapidly from about 0.5 to 4.0, which indicates that the model is trying to find a better generation strategy, thus increasing losses. Between 1500 and 2500 epochs, the loss value fluctuates greatly between 1.5 and 3.5, which means that generator faces challenges in adversarial training against discriminator, resulting in large fluctuations in the quality of generated samples. After 2500 epochs, the loss value shows a certain tendency to oscillate but generally stabilizes between about 2.0 and 2.5, achieving a relatively good result, which depicts that the model has reached a state of equilibrium where adversarial training makes it difficult for generator to improve performance further.

The loss curve and accuracy curve of discriminator for the training process are shown in Fig. 6. Similarly with the generator, it can be observed that after 2500 epochs, the loss value stabilizes, staying between 0.1 and 0.4. As shown in the accuracy curve, at the beginning of training (around the first 500 epochs), the accuracy rate rises from about 0.5 to close to 1.0 sharply, indicating that discriminator quickly learns to

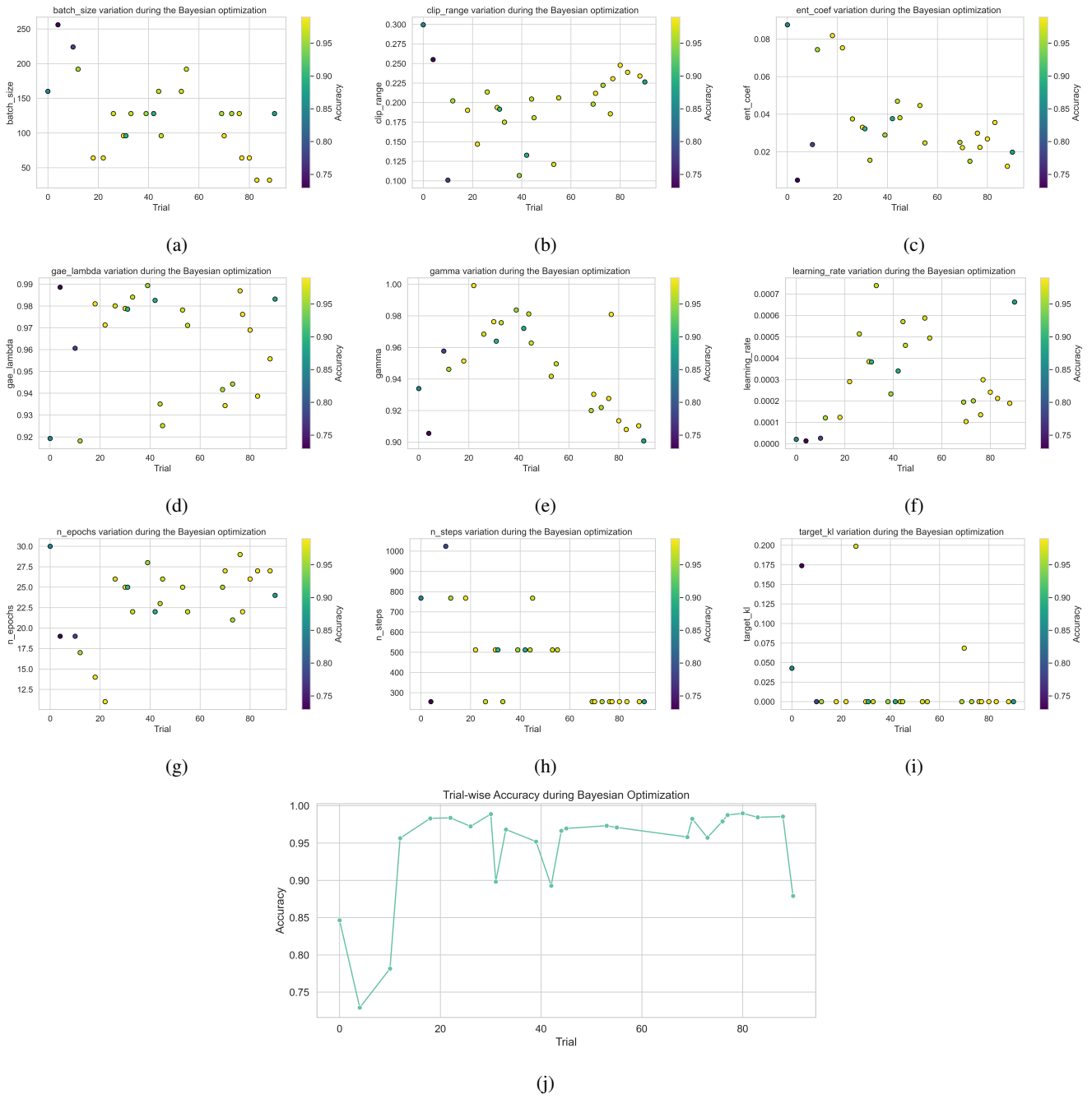


Fig. 4: Trial-wise Accuracy during Bayesian Optimization

distinguish between real data and generated data. Between 500 and 2500 epochs, the accuracy rate fluctuates greatly between 0.6 and 1.0, which is a common phenomenon in adversarial training, where generator and discriminator compete against each other. After 2500 epochs, the accuracy rate is stable around 0.95, showing that the discriminator improves the performance significantly with training, thus can accurately distinguish between real data and generated data.

TABLE VII lists the numbers of samples for each category before and after the expansion by C-GAN. It can be observed that the original sample sizes of BFA, Web, Botnet, and U2R are insufficient, while data augmentation increases these samples in the training set intentionally and significantly, with ratios ranging from $30.80\times$ to $288.75\times$. However, data of testing set has not changed, which helps to maintain the consistency of testing set and provide a stable standard for model evaluation. In all, through data augmentation, our proposed model is expected to perform better in processing unbalanced dataset and improve the identification accuracy of various types of attacks, especially for under-represented attack categories.

In order to observe how the dropout regularization mechanism improves the quality of generated samples in C-GAN, we design a group of ablation experiment. In the experiment,

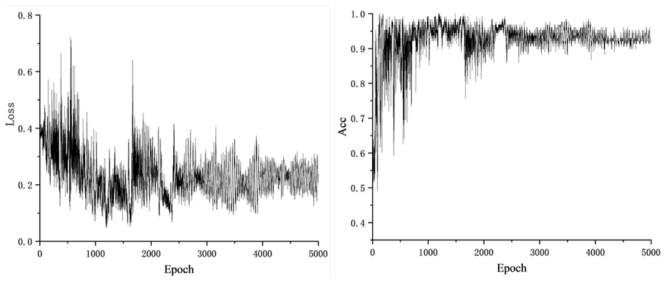


Fig. 6: Loss Curve of Discriminator

TABLE VII: Number of Samples before and after C-GAN

Attack Category	Raw Dataset		Augmented Dataset	
	Training Set	Testing Set	Training Set	Testing Set
Normal	47,897	20,527	56,210	20,527
DDoS	85,359	36,583	85,359	36,583
Probe	68,690	29,439	68,690	29,439
DoS	37,531	16,085	74,817	16,085
BFA	984	421	31294	421
Web	134	58	7,319	58
Botnet	115	49	5,784	49
U2R	12	5	3,477	5

we apply C-GAN to InSDN dataset, vary the dropout layer parameter of the discriminator from 0 to 0.2, and draw the UMAP (Uniform Manifold Approximation and Projection) dimension reduction diagrams of the generated data and original data to prove the impact of dropout layer on the quality of the generated data, which are shown in Fig. 7. It can be observed that in the absence of dropout, the distribution of the generated samples is relatively loose, which may result in a considerable distance between the generated samples and the actual samples. After using dropout, the distribution of the generated samples becomes more concentrated, and the overlapping areas with the real samples in multiple categories increase, such as Botnet, DDoS, and Probe, indicating that the quality of the generated samples has improved. In a conclusion, dropout helps reduce model overfitting and enhance the model's generalization ability.

VI. Performance Evaluations of Attack Classification

A. Ablation Experiments on Binary Classification

As GCB-PPO2 model uses our previously proposed CNNA-BiLSTM as the shared network, then the ablation experiments to compare GCB-PPO2 with CNN, CNN-LSTM, CNN-BiLSTM as well as CNNA-BiLSTM are essential. More specifically, we conduct comprehensive ablation studies through progressive architectural comparisons and generative enhancement validation. The architectural evolution analysis begins with a baseline CNN model using pure convolutional layers, subsequently enhanced by incorporating unidirectional temporal dependencies in CNN-LSTM, then upgraded to bidirectional context modeling in CNN-BiLSTM, and ultimately integrated with attention mechanisms to form the CNNA-BiLSTM variant. To specifically isolate the impact of C-GAN, we further compare the full GCB-PPO2 model (leveraging C-GAN synthesized under-represented class samples) against its ablated version CNNA-BiLSTM-PPO2 trained solely on original data. This dual-axis evaluation strategy enables systematic quantification of both architectural innovations and data augmentation effects, particularly in addressing class imbalance challenges.

TABLE VIII: Performance of Ablation Experiments on Binary Classification

Model	Accuracy (%)	Recall (%)	Precision (%)	F1 (%)
CNN	99.52	99.43	99.33	99.37
CNN-LSTM	99.50	99.51	99.46	99.48
CNN-BiLSTM	99.64	99.55	99.52	99.53
CNNA-BiLSTM	99.86	99.70	99.53	99.63
CNNA-BiLSTM-PPO2	99.89	**99.89**	98.07	98.31
GCB-PPO2	**99.92**	99.84	**99.95**	**99.89**

TABLE IX: Training Time of Ablation Experiments on Binary Classification

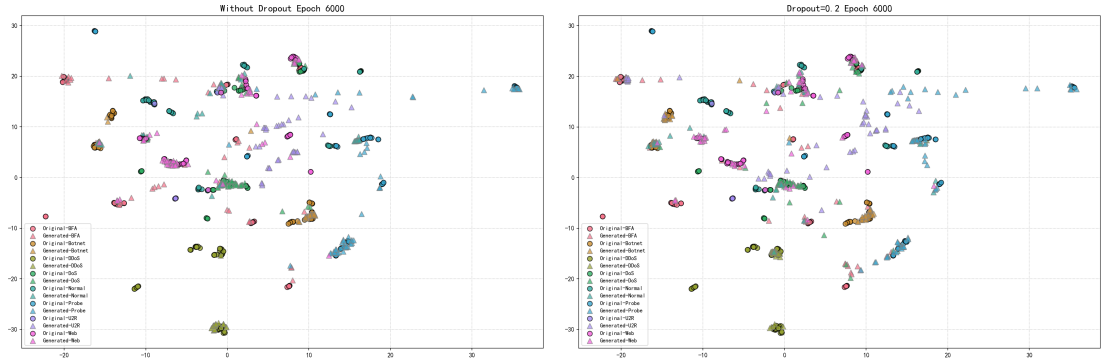
Model	Training Time (s)
CNN	714
CNN-LSTM	5842
CNN-BiLSTM	4331
CNNA-BiLSTM	3736
CNNA-BiLSTM-PPO2	2840
GCB-PPO2	2904

The binary classification effect of six models is shown in TABLE VIII. It can be concluded that GCB-PPO2 model achieves the best performance among six models on accuracy, precision, and F1. GCB-PPO2 model performs better than CNNA-BiLSTM-PPO2 on accuracy, precision, and F1, which reflects the advantage of C-GAN, that is, it can alleviate the problem of class imbalance by generating diversified attack traffic samples and improve the ability to identify under-represented attack categories.

In order to better demonstrate the performance of six models, the confusion matrixes are drawn and shown in Fig. 8. It can be observed that the true positive rate of anomaly packets by GCB-PPO2 model is 1.0, indicating that all samples which are malicious traffics are correctly identified as attacks. As a result, GCB-PPO2 model can detect all samples of attacks sensitively.

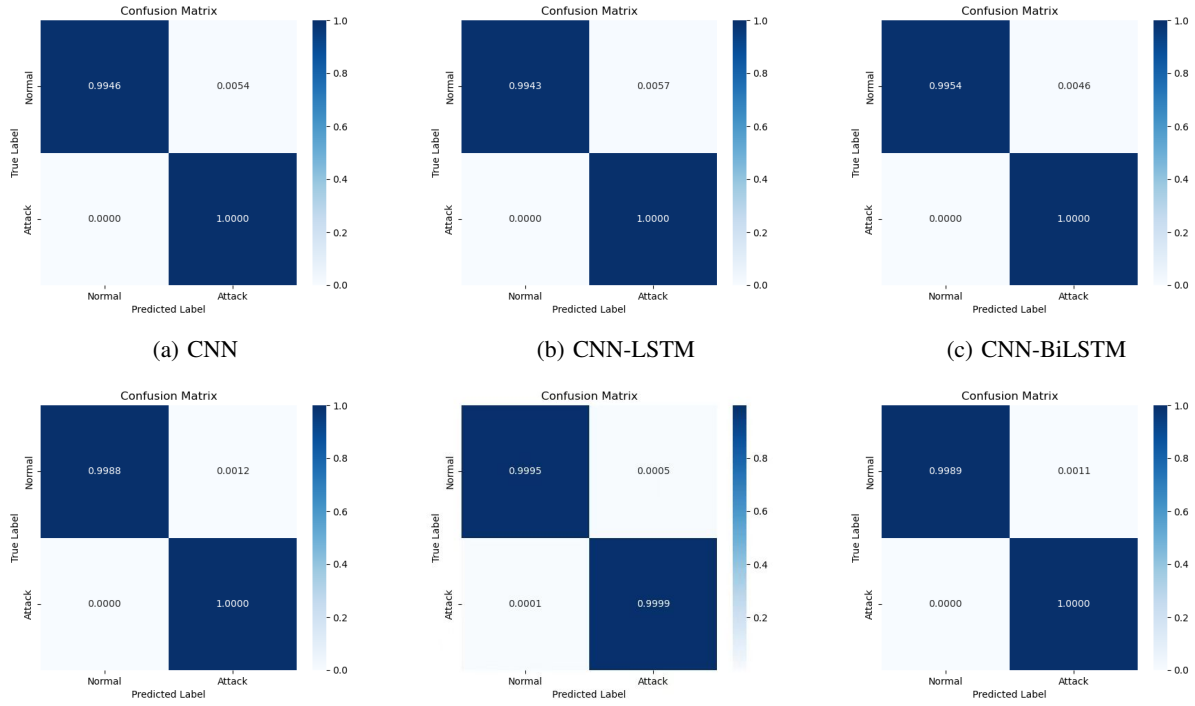
As models become more complex, the training time to be consumed is expected to increase in theory. However, PPO2 framework is introduced in this paper to speed up the convergence speed. The training time of six models is summarized in TABLE IX. It can be observed that the training time of the pure CNN model is only 714 seconds, which takes the shortest time among these models. However, as models become more complex, the training time required increases rapidly, and the longest time reaches 5842 seconds by CNN-LSTM model. Compared with other models, GCB-PPO2 model has the most complicated structure, while its training time is only higher than that of CNN and CNNA-BiLSTM-PPO2, and obviously less than that of CNN-LSTM, CNN-BiLSTM, and CNNA-BiLSTM models by 50.29%, 32.95%, and 22.27%, respectively. Moreover, GCB-PPO2 model consumes only 2.25% more training time than CNNA-BiLSTM-PPO2, which is negligible.

The reason why our proposed GCB-PPO2 model effectively reduces the required training time than CNN-LSTM based deep learning models can be explained as follows. Firstly, GCB-PPO2 model employs CNNA-BiLSTM as a shared net-



(a) (b)

Fig. 7: UMAP Diagram with and without Dropout Layer of C-GAN



(a) CNN (b) CNN-LSTM (c) CNN-BiLSTM

(d) CNNA-BiLSTM (e) CNNA-BiLSTM-PPO2 (f) GCB-PPO2

Fig. 8: Confusion Matrixes of Ablation Experiments on Binary Classification

work among multiple tasks and data streams, thus reducing computational redundancy as the same feature extraction layer does not need to be computed for multiple times. Secondly, GCB-PPO2 model utilizes the parallel computing techniques, such as N parallel agents collecting T steps of data in each iteration, to speed up the training process. Thirdly, GCB-PPO2 model adopts the sample reuse techniques such as experience pool to reduce the need for collecting new data and optimize the training time.

B. Ablation Experiments on Multiple Classification

TABLE X lists the precision, recall, and F1 of each traffic type when applying our proposed GCB-PPO2 model. It can be observed that GCB-PPO2 model achieves good results on each indicator for each category, including under-represented attack categories, i.e., BFA, Botnet, Web, and U2R attacks. More specifically, the precision, recall, and F1 of these minority classes are more than 86.36%, 84.28% and 85.91%, respectively. In other words, GCB-PPO2 model can identify all kinds of intrusion detection attacks with high performance.

On the basis of binary classification, we conduct ablation experiments of multiple classification to compare GCB-PPO2 model with five models mentioned in the last subsection. The simulation results are shown in TABLE XI. As depicted from the table, GCB-PPO2 model achieves the best performance among six models, with the accuracy being 99.01%, which is higher than that of other models by up to 2.42%.

The training time of six models on multiple classification is summarized in TABLE XII. When compared with CNN-LSTM based deep learning models, GCB-PPO2 model still

TABLE X: Performance of Traffic Types by GCB-PPO2 Model

Traffic Type	Precision (%)	Recall (%)	F1 (%)
Normal	99.26	99.14	99.19
DDoS	98.89	99.27	99.07
Probe	97.43	97.21	97.31
DoS	97.26	98.98	98.11
BFA	95.62	94.37	94.99
Botnet	87.62	84.28	85.91
Web	86.54	89.23	87.86
U2R	86.36	86.57	86.46

consumes much less training time. More specifically, its training time is less than that of CNN-LSTM, CNN-BiLSTM, and CNNA-BiLSTM models by 50.86%, 33.72%, and 29.73%, respectively. Simultaneously, GCB-PPO2 model consumes only 9.62% more training time than CNNA-BiLSTM-PPO2, which is acceptable. In a conclusion, GCB-PPO2 model consumes much less training time than traditional deep learning models effectively by at least 22.27% and 29.73% in binary and multiple classification scenarios, respectively, demonstrating improved computational efficiency. It can be attributed to the online learning ability of GCB-PPO2 model. Moreover, we have also integrated Bayesian optimization to fine-tune PPO2 hyper-parameters, further improving training stability and reducing computation cost.

In order to analyze the simulation results more deeply, we draw confusion matrixes for six models, which are shown in Fig. 9, and we list a table to display the F1 score of six models on each traffic type, which is shown in TABLE XIII. It can be observed from the confusion matrixes that GCB-PPO2 model correctly detects all the U2R attack samples, while other four deep learning models cannot detect this type at all. Moreover, GCB-PPO2 model correctly detects 88% of Web attack samples, while CNN, CNN-LSTM, and CNN-BiLSTM models cannot detect Web traffics at all, and CNNA-BiLSTM-PPO2 model only detects 7% of Web traffics.

As shown in the confusion matrixes, CNN, CNN-LSTM, CNN-BiLSTM, and CNNA-BiLSTM-PPO2 models mistake Web attacks as BFA traffics, and the former three models mistake U2R attacks as normal traffics. On one hand, Web attacks try different attack tactics (such as SQL injection, cross-site scripting) to find a system's weakness, while BFA attacks try all possible password combinations to find the right cipher. Both attacks exhibit persistent attempts to access system resources. When models detect a large number of repeated login or access attempts, they may classify these actions as BFA attacks. On the other hand, attackers of U2R attacks perform malicious actions by obtaining system permissions with the characteristics of small number of packets, large packet size, low traffic fluctuations and strong purpose of access, which are similar to normal user behaviors, so it is easy to be mistaken as normal traffics.

TABLE XIII demonstrates that the GCB-PPO2 model excels in detecting all the traffic types, achieving an F1 score of at least 85.91%. Notably, it performs exceptionally well in identifying all the under-represented attack categories, including BFA, Botnet, U2R, and Web attacks, with F1 scores of 94.99%, 85.91%, 86.47%, and 87.86%, respectively. On

TABLE XI: Performance of Ablation Experiments on Multiple Classification

Model	Accuracy (%)	Recall (%)	Precision (%)	F1 (%)
CNN	96.59	96.58	96.58	96.53
CNN-LSTM	96.61	96.24	96.62	96.42
CNN-BiLSTM	96.66	96.24	96.66	96.42
CNNA-BiLSTM	96.66	96.66	96.66	96.62
CNNA-BiLSTM-PPO2	98.57	98.57	98.07	98.31
GCB-PPO2	**99.01**	**99.01**	**99.00**	**98.98**

TABLE XII: Training Time of Ablation Experiments on Multiple Classification

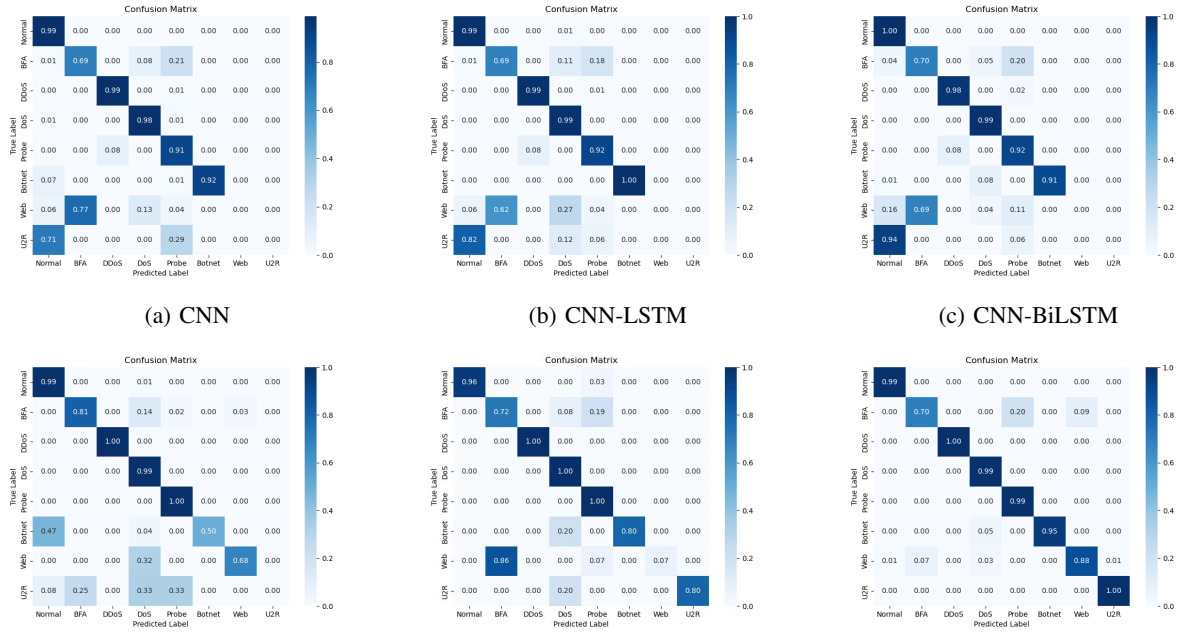
Model	Training Time (s)
CNN	700
CNN-LSTM	5800
CNN-BiLSTM	4300
CNNA-BiLSTM	3700
CNNA-BiLSTM-PPO2	2600
GCB-PPO2	2850

the contrary, CNN, CNN-LSTM, and CNN-BiLSTM fail to detect U2R and Web attacks. CNNA-BiLSTM shows limited success, obtaining F1 scores of 0% and 48.91% for U2R and Botnet, respectively. While the CNNA-BiLSTM-PPO2 effectively detects most traffic types with F1 scores of more than 77.80%, its performance on Web attacks is notably weak, with an F1 score of only 12.50%.

The sample sizes for BFA, Web, Botnet, and U2R are only 1405, 192, 164 and 17, respectively, accounting for merely 0.44%, 0.06%, 0.05% and 0.004% of all samples. This severe class imbalance poses a significant challenge to intrusion detection models. Five models for comparison do not solve data imbalance problem effectively. Due to the limited number of minority class samples, these models cannot fully learn the features of abnormal traffics during training, resulting in the lack of sufficient recognition ability when facing new malicious flows. As a result, it is difficult for these five models to identify BFA, Web, Botnet, and U2R attacks.

In order to investigate classification results more intuitively, we use t-distributed Stochastic Neighbor Embedding (t-SNE) method to draw a graph to visualize the policy latent space of GCB-PPO2 model, as shown in Fig. 10. The data points in this figure are represented by t-SNE after dimensionality reduction, with the original high-dimensional feature space mapped to a two-dimensional space, helpful for observing the distribution of different categories of data more conveniently. In general, different categories of data still maintain good clustering form after dimensionality reduction, indicating that GCB-PPO2 model can distinguish different types of network traffic effectively.

More specifically, the following conclusions can be derived from this figure. (1) Normal, DDoS, and DoS traffics form a certain clustering for each, indicating that these traffics are relatively stable and can be clearly distinguished from other attack categories. (2) The distribution of Probe attacks is wide, indicating that this traffic mode is diverse. When traffic of Probe attacks is running at a low rate, it is difficult to detect directly from traffic patterns. (3) There are few data points for Web, U2R, Botnet, and BFA attacks. The distribution is scattered, and no obvious clustering is formed. As the samples



(a) CNN (b) CNN-LSTM (c) CNN-BiLSTM

(d) CNNA-BiLSTM (e) CNNA-BiLSTM-PPO2 (f) GCB-PPO2

Fig. 9: Confusion Matrixes of Ablation Experiments on Multiple Classification

TABLE XIII: F1 of Traffic Types in Ablation Experiments

F1 (%) / Traffic Type	CNN	CNN-LSTM	CNN-BiLSTM	CNNA-BiLSTM	CNNA-BiLSTM-PPO2	GCB-PPO2
Normal	99.41	99.32	**99.62**	99.24	98.03	99.20
DDoS	96.12	96.10	95.98	99.97	**100.00**	99.08
Probe	94.38	94.52	94.36	**99.70**	98.45	97.27
DoS	98.74	98.87	99.32	98.58	**99.49**	98.11
BFA	73.19	72.20	74.70	82.00	77.80	**94.99**
Web	0	0	0	64.70	12.50	**87.86**
Botnet	80.38	**90.32**	87.46	48.91	82.10	85.91
U2R	0	0	0	0	80.00	**86.47**

Fig. 10: Policy Latent Space

of Web, U2R, Botnet, and BFA are very few, their detection effects are limited.

C. Contrast Experiments with Traditional Algorithms

In this subsection, we compare our proposed GCB-PPO2 model with traditional machine learning models on accuracy, precision, recall, and F1 to observe the difference between machine learning and DRL models on the performance of intrusion detection. Machine learning models for comparison include Decision Tree (DT), Support Vector Machine (SVM), Random Forest (RF), Logistic Regression (LR), Multi-Layer Perceptron (MLP), K-Nearest Neighbor (KNN), Stochastic Gradient Descent (SGD), and Adaptive Boosting (AdaBoost). The simulation results are shown in TABLE XIV.

It can be concluded that the performance difference between traditional machine learning and DRL is not obvious when applying to binary classification tasks, while there is a huge gap when conducting multiple classification tasks. In multiple classification scenario, compared with other models, GCB-PPO2 model improves accuracy, precision, recall, and F1 by at least 5.68%, 4.53%, 5.85%, and 5.17%, respectively. This is because in multiple classification problems, the increase of categories is usually accomplished by the increase of problem complexity. Due to the multi-layer network structures and activation functions of deep learning, GCB-PPO2 model has higher expressiveness to capture category differences and more powerful handling ability to solve nonlinear mapping problems, resulting in the best performance when applying to

TABLE XIV: Contrast Experiments with Traditional Algorithms

Model	Binary Classification				Multi-Classification			
	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)
Decision Tree	98.78	99.54	99.64	99.58	92.10	90.26	91.05	90.65
Support Vector Machine	99.01	**100.00**	99.31	99.63	86.98	90.94	76.32	82.99
Random Forest	99.37	99.81	**99.91**	99.85	82.04	84.75	82.05	83.37
Logistic Regression	99.80	99.03	97.32	98.16	59.68	60.71	59.74	60.22
Multi-Layer Perceptron	**100.00**	99.27	96.19	97.70	93.16	94.47	93.16	93.81
K-Nearest Neighbor	97.24	99.61	98.75	99.17	93.33	93.17	84.04	88.36
Stochastic Gradient Descent	99.85	99.38	99.24	99.30	57.27	75.91	52.78	62.26
Adaptive Boosting	99.79	99.83	**99.91**	99.86	78.57	71.49	78.25	74.71
GCB-PPO2	99.92	99.95	99.84	**99.89**	**99.01**	**99.00**	**99.01**	**98.98**

complex classification boundary problems.

On the other hand, DT, MLP, and KNN demonstrate the effectiveness in multiple classification, with the accuracy of more than 92.10%. It can be attributed to the following three reasons. (1) These models have nonlinear modeling capability. For instance, DT directly constructs nonlinear decision boundaries through tree splitting. (2) These models can process high dimensional data effectively. For instance, MLP fuses multi-dimensional features through the fully connected layer. (3) These models are robust to noise. For instance, KNN adjusts the number of neighbours by the K value to balance noise effects. As a result, these three models achieve competitive performance, while remain substantially inferior to GCB-PPO2 model.

Secondly, the performance of SVM, RF, and AdaBoost is acceptable. However, these three methods have limitations on model structures and defects on data adaptation. For instance, SVM requires manual selection of kernel functions, and RF voting mechanism fails when classes are unbalanced. As a result, they do not perform as well as the first three models. Finally, the performance metrics of LG and SGD are lowest among all the models, as both of them are suitable for dealing with linear data, leading to poor performance in dealing with nonlinear and multiple classification problems.

D. Contrast Experiments with State-of-the-Art Algorithms

In order to further demonstrate the superiority of our proposed GCB-PPO2 model, we conduct experiments to compare it with other state-of-the-art algorithms on multiple classification scenarios for InSDN dataset. The compared models include the model proposed in [60], D-GSAGE-MARC [61], AE-CNN [62], J48 [63], CNN-MLP [64], DCGAN [65], FELIDS [66], and ByteSGAN [67]. These studies were published between 2021 and 2025, representing recent advancements in this filed. The comparison results are shown in TABLE XV[1]. The results demonstrate that the GCB-PPO2 model consistently outperforms the other eight models, achieving improvements of up to 15.56%, 16.27%, 15.11%, and 15.68% in accuracy, recall, precision, and F1, respectively.

As shown in TABLE XV , CNN-RF ,CNN-MLP and FELIDS exhibit the lowest performance among all models across the four metrics. On one hand, CNN-RF combines the advantages of deep learning and ensemble learning, but the collaborative efficiency of the two may be insufficient. CNN is good at extracting local spatial features, while RF relies on global feature classification. Using the output of CNN as the input of RF may lead to incomplete feature representation, thereby resulting in the degradation of intrusion detection performance. On the other hand, CNN-MLP and FELIDS belong to federated learning based IDSs, which need to be trained on multiple nodes. However, data distribution on each node may be uneven, making it difficult for these models to learn features of global data, and further affecting the generalization ability and classification accuracy of models. Moreover, technologies such as differential privacy may be introduced to protect sensitive data, which may negatively affect model performance and reduce classification accuracy. As a result, the performance of federated learning models in intrusion detection tasks may be inferior to that of centralized learning models.

[1]The precision of D-GSAGE-MARC is not provided.

TABLE XV: Contrast Experiments with State-of-the-Art Algorithms (2021-2025) on Multiple Classification

Model	Accuracy (%)	Recall (%)	Precision (%)	F1 (%)
[60] (2025)	98.42	96.96	96.30	97.08
D-GSAGE-MARC [61] (2025)	97.11	97.11	-	97.05
AE-CNN [62] (2024)	95.27	95.67	94.92	95.30
CNN-RF [63] (2024)	83.45	82.74	83.89	83.30
CNN-MLP [64] (2023)	87.73	88.69	87.12	87.85
DCGAN [65] (2022)	93.88	94.11	93.70	93.91
FELIDS [66] (2022)	89.14	89.01	89.22	89.12
ByteSGAN [67] (2021)	92.95	92.36	93.41	92.89
GCB-PPO2 (Ours)	**99.01**	**99.01**	**99.00**	**98.98**

E. Cross-Domain Experiments on NSL-KDD Dataset

In this study, C-GAN is used as a data enhancement tool aimed at improving the fairness of our proposed GCB-PPO2 model for classifying different traffic types by generating more diverse samples of attacks. Specifically, C-GAN helps the model avoid the bias to common attack types in the training process by generating samples of under-represented attack types, thus improving the model's ability to identify various traffic types. Meanwhile, we should discuss the potential bias introduced by data augmentation techniques, and confirm whether C-GAN overfits specific types of attacks. In this subsection, we apply GCB-PPO2 model to another dataset named "NSL-KDD" to test the performance of the model on a different dataset as well as to explore the applicability in various network conditions and environments.

Similarly, both binary and multiple classification experiments are designed to evaluate the performance of the GCB-PPO2 model in distinguishing normals traffic from abnormal attacks, and to verify the generalization ability of the model

TABLE XVI: Cross-Domain Experiments on NSL-KDD Dataset by GCB-PPO2 Model

Experiment	Accuracy (%)	Recall (%)	Precision (%)	F1 (%)
Binary Classification	99.66	99.61	99.70	99.65
Multiple Classification	98.65	98.65	98.69	98.66

TABLE XVII: Performance of Traffic Types on NSL-KDD Dataset by GCB-PPO2 Model

Traffic Type \ Metric	Precision (%)	Recall (%)	F1 (%)
Normal	99.17	97.94	98.55
DoS	99.68	99.77	99.73
Probe	99.75	99.96	99.86
R2L	93.23	97.63	95.38
U2R	97.81	89.50	93.47

for complex traffic types. The corresponding accuracy, recall, precision, and F1 are shown in TABLE XVI. It can be concluded that GCB-PPO2 model performs well in binary classification tasks, with all the four metrics reaching above 99.61%. In multi-classification tasks, the performance of the model decreases slightly, while the overall level still remains high, as all the four metrics are higher than 98.86%, verifying its robustness in complex network environment.

On that basis, we measure the fine-grained classification performance of GCB-PPO2 model on NSL-KDD dataset, which is summarized in TABLE XVII. It can be concluded that the F1 of GCB-PPO2 model for all the five types of traffics is above 93.47%, further verifying its multi-classification capability. The simulation results verify that GCB-PPO2 model still achieves a high accuracy on a different dataset, indicating that C-GAN successfully maintains diversity and representativeness when generating samples, and does not lead to bias in specific attack types. Moreover, the results also indicates that GCB-PPO2 model can adapt to different data distributions and has strong generalization ability.

F. Analysis Experiments of Feature Importance

In order to analyze the feature importance, we import SHAP (Shapley Additive exPlanations) tool to help explain the decision-making process of our proposed GCB-PPO2 model. The core idea of SHAP is to calculate the importance of features by comparing the effect of the addition and removal of features on the model output. In terms of each prediction, SHAP calculates the Shapley value for each feature, representing the contribution of that feature to the final prediction results. Based on the results, Matplotlib is used to draw a bar graph showing the importance of each feature, as shown in Fig. 11.

It can be observed that the following characteristics are of significant importance to the final classification results: Bwd Pkt Len Min (minimum reverse packet length), Pkt Len Min (minimum packet length), Bwd IAT Max (maximum reverse arrival time interval), Bwd IAT Std (reverse arrival time standard deviation), and Flow IAT Mean (average flow arrival time). The greater the absolute value of a feature's importance score, the stronger its influence on the GCB-PPO2 model's decision-making. Based on the experimental results, the following conclusions can be derived.

- Reverse flow characteristics such as Bwd Pkt Len Min, Bwd IAT Max, and Bwd IAT Std are particularly effective for detecting attacks.
- Packet length and traffic time interval (IAT based features) contribute significantly to classification. DDoS may use small packets, Botnet may have a fixed communication interval, which can be distinguished by packet length and traffic time interval, respectively.
- The impact of forward flow characteristics is relatively low, such as Fwd Header Len and Fwd Pkts/s.

G. Robustness Experiments to Adversarial Attacks

In this subsection, we design experiments by using Fast Gradient Sign Method (FGSM) to observe the robustness of our proposed GCB-PPO2 model to adversarial attacks. FGSM is one of the most commonly used white box attack methods, which generates the disturbance by calculating the gradient of the loss function. The process of experiments includes: (1) load the pre-training model, (2) generate counter-samples, (3) test in batches, and (4) output the results.

- Generate Counter-Samples: We set different $epsilon$ values to simulate different disturbance strengths. The $epsilon$ values set in the experiments are $[0.001, 0.01, 0.1]$, covering the small disturbance to large disturbance, and can comprehensively evaluate the performance of GCB-PPO2 model under different attack intensities.
- Test in Batches: The reason why the tests are conducted in batches is to avoid memory deficits. In the experiments, we reduce batch sizes to 8 and regularly clean up video memory to better apply to large data or resource-constrained environments.
- Output Results: In the experiments, we calculate accuracy, precision, recall, and F1 in different scenarios with different $epsilon$ values. Moreover, we not only test the overall performance (shown in TABLE XVIII), but also analyze performance differences across different attack classes (shown in TABLE XIX).

As shown in TABLE XVIII, all indexes decrease significantly with the increase of anti-disturbance intensity. When $epsilon = 0.001$ and $epsilon = 0.01$, the performance of GCB-PPO2 is acceptable (as all metrics are beyond 82.32%), which means that the model has a certain tolerance for low intensity perturbations. While when $epsilon = 0.1$, the model performance is close to random guess level (as accuracy, recall, and F1 are below 50%), indicating that the model almost fails under high intensity disturbance.

As shown in TABLE XIX, there are significant differences in the detection robustness of different traffic types. On one hand, in terms of Normal, DDoS, DoS, and Probe classes, GCB-PPO2 model is robust to low intensity perturbations ($epsilon = 0.001$ and $epsilon = 0.01$), as $F1 > 71.24\%$. While the performance deceases significantly at high intensity perturbations ($epsilon = 0.1$). While these attacks are common in SDN, their inclusion in training ensures model robustness against high-volume threats. However, our focus remains on detecting under-represented attacks. On the other

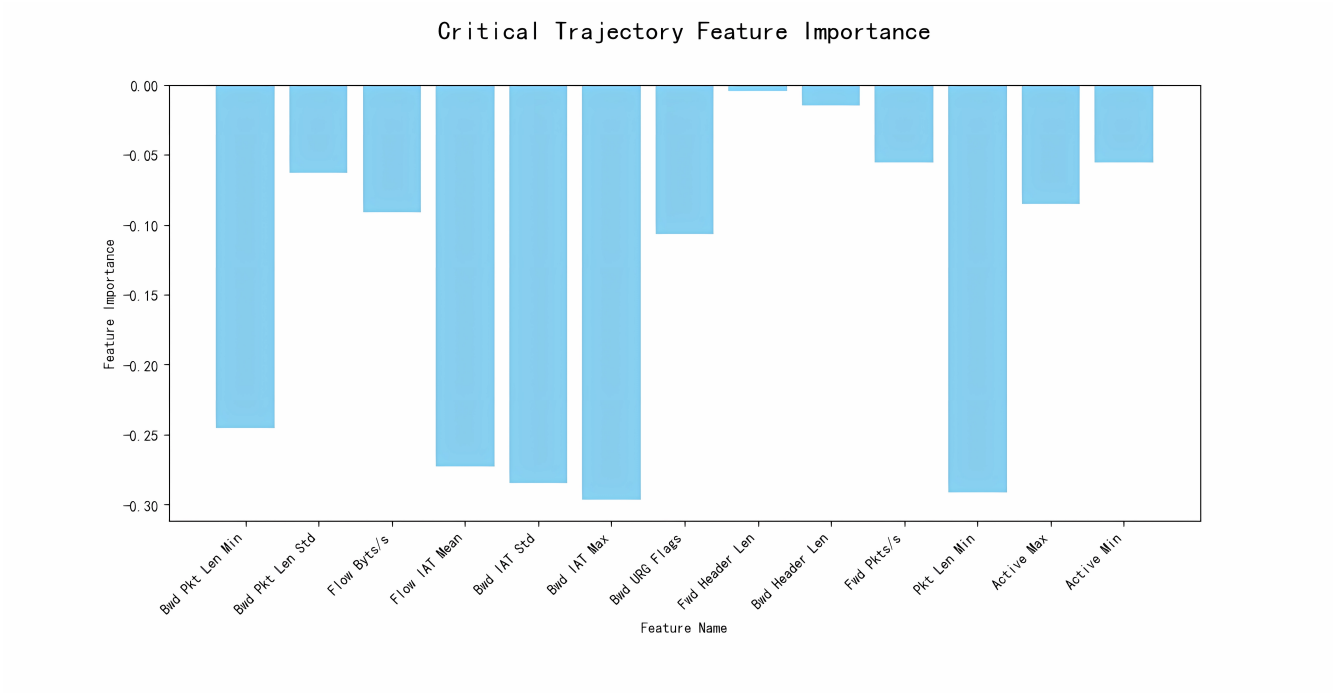


Fig. 11: Bar Graph of Feature Importance by GCB-PPO2 Model

TABLE XVIII: Robustness Experiments by GCB-PPO2 Model

Metric \ *epsilon* Values	Original Model	$epsilon = 0.001$	$epsilon = 0.01$	$epsilon = 0.1$
Accuracy (%)	99.01	86.58	82.32	45.63
Recall (%)	99.01	86.58	82.32	45.63
Precision (%)	99.00	90.26	86.74	74.55
F1 (%)	98.98	86.99	82.66	46.68

TABLE XIX: Performance of Traffic Types under Robustness Experiments by GCB-PPO2 Model

Traffic Type \ F1 (%)	Original Model	$epsilon = 0.001$	$epsilon = 0.01$	$epsilon = 0.1$
Normal	99.20	74.66	71.24	43.05
DDoS	99.08	98.00	97.92	44.81
Probe	97.27	87.63	78.59	49.95
DoS	98.11	78.67	72.23	51.11
BFA	94.99	25.84	16.60	0
Web	87.86	0	0	0
Botnet	85.91	7.84	0	0
U2R	86.47	0	0	0

hand, in terms of BFA, Web, U2R, and Botnet classes, F1 score has plummeted to near 0 even under weak perturbations. As mentioned before, these four traffic types, categorized as low-prevalence attacks, exhibit heightened vulnerability to adversarial perturbations, primarily stemming from their strong dependence on few key features, insufficient sample size, and marginalization of feature distribution. In counter-sample attacks, noise can obscure these already obscure features, causing the model to fail to recognize them correctly. In a conclusion, the current GCB-PPO2 model can tolerate low intensity perturbations, while the robustness of high intensity perturbations is insufficient, and the difference between categories is significant.

VII. Conclusions and Future Works

In this paper, we present GCB-PPO2, a deep reinforcement learning based hybrid intrusion detection system for SDN environments that synergistically optimizes classification performance and training efficiency through two integrated innovations: (1) C-GAN based adversarial generation of under-represented attack synthesis to counteract class imbalance, and (2) enhanced PPO2 reinforcement learning incorporating our prior CNNA-BiLSTM architecture for spatio-temporal feature fusion. Experimental evaluations demonstrate that GCB-PPO2 model achieves the accuracy of 99.92% and 99.01% in binary and multiple classification scenarios, respectively, and elevates the F1 of under-represented attack detection to over 85.91%. Moreover, GCB-PPO2 model proves cross-domain efficacy with 98.65% accuracy on NSL-KDD benchmark, and achieves

22.27% faster convergence than conventional hybrid models.

To strengthen the practical applicability and ethical accountability of GCB-PPO2 model, we plan to integrate this model as a component or module of the controller(s). Future research will focus on following three aspects: technical optimization, theoretical optimization, and scenario expansion.

- Technical Optimization: Transferring the knowledge of GCB-PPO2 to a lightweight model to streamline training complexity and apply to resource limited SDN controllers. Integrating the adversarial perturbation penalty term into the reward function of PPO2 to enhance the robustness to adversarial attacks.
- Theoretical Optimization: Using the causal reasoning model to identify the causal path between the attack features and the classification results to improve the interpretability.
- Scenario Expansion: Designing an unsupervised anomaly scoring module and training it jointly with the existing supervised learning to deal with zero-day attacks.

To improve the performance of GCB-PPO2 model, both the data augmentation strategy and the intrusion detection model can be enhanced. While C-GAN effectively addresses class imbalance, integrating it with other techniques such as SMOTE and contrastive learning could further diversify synthetic samples, especially for extremely low frequency categories (e.g., U2R attacks). Exploring transformer-based modules or Graph Neural Networks (GNNs) could better capture hierarchical dependencies in network traffic. For example, a spatial-temporal transformer might replace the CNNA-BiLSTM to model long-range correlations in attack patterns. These advancements aim to push the performance boundaries of SDN IDS in evolving threat landscapes.

VIII. Acknowledgements

This research is founded by National Natural Science Foundation of China Youth Fund Program (62102241).

References

[1] Rohit Kumar, U. Venkanna, and Vivek Tiwari. Optimized traffic engineering in software defined wireless network based iot (sdwn-iot): State-of-the-art, research opportunities and challenges. *COMPUTER SCIENCE REVIEW*, 49, AUG 2023.

[2] Chunlin Li, Kun Jiang, and Youlong Luo. Dynamic placement of multiple controllers based on sdn and allocation of computational resources based on heuristic ant colony algorithm. *KNOWLEDGE-BASED SYSTEMS*, 241, APR 6 2022.

[3] Jue Chen, Yu-Jie Xiong, Xihe Qiu, Dun He, Hanmin Yin, and Changwei Xiao. A cross entropy based approach to minimum propagation latency for controller placement in software defined network. *COMPUTER COMMUNICATIONS*, 191:133–144, JUL 1 2022.

[4] Zaid Mustafa, Rashid Amin, Hamza Aldabbas, and Naeem Ahmed. Intrusion detection systems for software-defined networks: a comprehensive study on machine learning-based techniques. *CLUSTER COMPUTING-THE JOURNAL OF NETWORKS SOFTWARE TOOLS AND APPLICATIONS*, 27(7):9635–9661, OCT 2024.

[5] Yue Zhang, Chen Jue, Wanxiao Liu, and Yurui Ma. Gran: a sdn intrusion detection model based on graph attention network and residual learning. *COMPUTER JOURNAL*, 68(3):241–260, OCT 24 2024.

[6] Liyan Yang, Yubo Song, Shang Gao, Aiqun Hu, and Bin Xiao. Griffin: Real-time network intrusion detection system via ensemble of autoencoder in sdn. *IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT*, 19(3):2269–2281, SEP 2022.

[7] Amir Javadpour, Pedro Pinto, Forough Ja'fari, and Weizhe Zhang. Dmaidps: a distributed multi-agent intrusion detection and prevention system for cloud iot environments. *CLUSTER COMPUTING-THE JOURNAL OF NETWORKS SOFTWARE TOOLS AND APPLICATIONS*, 26(1):367–384, FEB 2023.

[8] Emmanuel Hooper. Intelligent detection and response strategies for complex attacks. *IEEE AEROSPACE AND ELECTRONIC SYSTEMS MAGAZINE*, 22(11):3–12, NOV 2007. 40th Annual IEEE International Carnahan Conference on Security Technology, Lexington, KY, OCT 16-19, 2006.

[9] Wathiq Laftah Al-Yaseen, Zulaiha Ali Othman, and Mohd Zakree Ahmad Nazri. Multi-level hybrid support vector machine and extreme learning machine based on modified k-means for intrusion detection system. *EXPERT SYSTEMS WITH APPLICATIONS*, 67:296–303, JAN 2017.

[10] Iftikhar Ahmad, Mohammad Basheri, Muhammad Javed Iqbal, and Aneel Rahim. Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection. *IEEE ACCESS*, 6:33789–33795, 2018.

[11] Named Haddad Pajouh, Reza Javidan, Raouf Khayami, Ali Dehghantanha, and Kim-kwang Raymond Choo. A two-layer dimension reduction and two-tier classification model for anomaly-based intrusion detection in iot backbone networks. *IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTING*, 7(2):314–323, APR-JUN 2019.

[12] Baswaraju Swathi, Soma Sekhar Kolisetty, G. Venkata Sivanarayana, and Srinivasa Rao Battula. Efficientnetv2-regnet: an effective deep learning framework for secure sdn based iot network. *CLUSTER COMPUTING-THE JOURNAL OF NETWORKS SOFTWARE TOOLS AND APPLICATIONS*, 27(8):10653–10670, NOV 2024.

[13] Zhendong Wang, Yaodi Liu, Daojing He, and Sammy Chan. Intrusion detection methods based on integrated deep learning model. *COMPUTERS & SECURITY*, 103, APR 2021.

[14] Meng Cui, Jue Chen, Xihe Qiu, Wenjing Lv, Haijun Qin, and Xinyu Zhang. Multi-class intrusion detection system in sdn based on hybrid bilstm model. *CLUSTER COMPUTING-THE JOURNAL OF NETWORKS SOFTWARE TOOLS AND APPLICATIONS*, 2024 MAY 2 2024.

[15] Mateusz Buda, Atsuto Maki, and Maciej A. Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *NEURAL NETWORKS*, 106:249–259, OCT 2018.

[16] Mahmoud Said Elsayed, Nhien-An Le-Khac, and Anca D. Jurcut. Insdn: A novel sdn intrusion dataset. *IEEE ACCESS*, 8:165263–165284, 2020.

[17] Chengqing Liang, Lei Liu, and Chen Liu. Multi-uav autonomous collision avoidance based on ppo-gic algorithm with cnn-lstm fusion network. *NEURAL NETWORKS*, 162:21–33, MAY 2023.

[18] Xiangyin Zhang, Hao Zong, and Weihuan Wu. Cooperative obstacle avoidance of unmanned system swarm via reinforcement learning under unknown environments. *IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT*, 74, 2025.

[19] Man Li, Shuangxing Deng, Huachun Zhou, and Yajuan Qin. A path selection scheme for detecting malicious behavior based on deep reinforcement learning in sdn/nfv-enabled network. *COMPUTER NETWORKS*, 236, NOV 2023.

[20] Boshra Darabi, Mozafar Bag-Mohammadi, and Mojtaba Karami. A micro reinforcement learning architecture for intrusion detection systems. *PATTERN RECOGNITION LETTERS*, 185:81–86, SEP 2024.

[21] Xiangyu Ma and Wei Shi. Aesmote: Adversarial reinforcement learning with smote for anomaly detection. *IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING*, 8(2):943–956, APR-JUN 2021.

[22] Weiping Wang, Junjiang Guo, Zhen Wang, Hao Wang, Jun Cheng, Chunyang Wang, Manman Yuan, Juergen Kurths, Xiong Luo, and Yang Gao. Abnormal flow detection in industrial control network based on deep reinforcement learning. *APPLIED MATHEMATICS AND COMPUTATION*, 409, NOV 15 2021. 3rd International Conference on Numerical Computations - Theory and Algorithms (NUMTA), Crotone, ITALY, JUN 15-21, 2019.

[23] Sunghwan Kim, Seunghyun Yoon, Jin-Hee Cho, Dong Seong Kim, Terrence J. Moore, Frederica Free-Nelson, and Hyuk Lim. Divergence: Deep reinforcement learning-based adaptive traffic inspection and moving target defense countermeasure framework. *IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT*, 19(4):4834–4846, DEC 2022.

[24] Noe M. Yungaicela-Naula, Cesar Vargas-Rosales, Jesus Arturo Perez-Diaz, and Diego Fernando Carrera. A flexible sdn-based framework for slow-rate ddos attack mitigation by reinforcement. *JOURNAL OF NETWORK AND COMPUTER APPLICATIONS*, 205, SEP 2022.

[25] Noe M. Yungaicela-Naula, Cesar Vargas-Rosales, and Jesus A. Perez-Diaz. Sdn/nfv-based framework for autonomous defense against slow-

rate ddos attacks by using reinforcement learning. *FUTURE GENERATION COMPUTER SYSTEMS-THE INTERNATIONAL JOURNAL OF ESCIENCE*, 149:637–649, DEC 2023.

[26] Weilong Chen, Shaoliang Zhang, Ruobing Xie, Feng Xia, Leyu Lin, Xinran Zhang, Yan Wang, and Yanru Zhang. Cippo: Contrastive imitation proximal policy optimization for recommendation based on reinforcement learning. *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*, 36(11):5753–5767, NOV 2024.

[27] Pierre-Luc Dallaire-Demers and Nathan Killoran. Quantum generative adversarial networks. *PHYSICAL REVIEW A*, 98(1), JUL 23 2018.

[28] Jiayu Wang, Xuehu Yan, Lintao Liu, Longlong Li, and Yongqiang Yu. Cttgan: traffic data synthesizing scheme based on conditional gan. *Sensors*, 22(14), 2022.

[29] Qingli Zeng and Farid Nait-Abdesselam. Leveraging human-in-the-loop machine learning and gan-synthesized data for intrusion detection in unmanned aerial vehicle networks. In *ICC 2024-IEEE International Conference on Communications*, pages 1557–1562. IEEE, 2024.

[30] Qingli Zeng and Farid Nait-Abdesselam. Enhancing uav network security: a human-in-the-loop and gan-based approach to intrusion detection. *IEEE Internet of Things Journal*, 2025.

[31] Center for Internet Security. The Mirai botnet: Threats and mitigations. CISecurity Blog, 2016.

[32] Amir Javadpour, Forough Jafari, Tarik Taleb, Mohammad Shojafar, and Chafika Benzaid. A comprehensive survey on cyber deception techniques to improve honeypot performance. *COMPUTERS & SECURITY*, 140, MAY 2024.

[33] Abbas Yazdinejadna, Reza M. Parizi, Ali Dehghantanha, and Mohammad S. Khan. A kangaroo-based intrusion detection system on software-defined networks. *COMPUTER NETWORKS*, 184, JAN 15 2021.

[34] Jaeik Cho, Taeshik Shon, Ken Choi, and Jongsub Moon. Dynamic learning model update of hybrid-classifiers for intrusion detection. *JOURNAL OF SUPERCOMPUTING*, 64(2):522–526, MAY 2013.

[35] Pynbianglut Hadem, Dilip Kumar Saikia, and Soumen Moulik. An sdn-based intrusion detection system using svm with selective logging for ip traceback. *COMPUTER NETWORKS*, 191, MAY 22 2021.

[36] Alan Y. P. Lee, Michael I. C. Wang, Chi-Hsiang Hung, and Charles H. P. Wen. Ps-ips: Deploying intrusion prevention system with machine learning on switch. *FUTURE GENERATION COMPUTER SYSTEMS-THE INTERNATIONAL JOURNAL OF ESCIENCE*, 152:333–342, MAR 2024.

[37] Zakaria Abou El Houda, Bouziane Brik, and Lyes Khoukhi. Ensemble learning for intrusion detection in sdn-based zero touch smart grid systems. In S Oteafy, E Bulut, and F Tschorsch, editors, *PROCEEDINGS OF THE 2022 47TH IEEE CONFERENCE ON LOCAL COMPUTER NETWORKS (LCN 2022)*, Conference on Local Computer Networks, pages 149–156. IEEE; IEEE Comp Soc, Tech Comm Comp Commun; TELUS, 2022. 47th IEEE Conference on Local Computer Networks (LCN), Edmonton, CANADA, SEP 26-29, 2022.

[38] Nisha Ahuja, Gaurav Singal, Debajyoti Mukhopadhyay, and Neeraj Kumar. Automated ddos attack detection in software defined networking. *JOURNAL OF NETWORK AND COMPUTER APPLICATIONS*, 187, AUG 1 2021.

[39] Chuanlong Yin, Yuefei Zhu, Jinlong Fei, and Xinzheng He. A deep learning approach for intrusion detection using recurrent neural networks. *IEEE ACCESS*, 5:21954–21961, 2017.

[40] Arwa Aldweesh, Abdelouahid Derhab, and Ahmed Z. Emam. Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues. *KNOWLEDGE-BASED SYSTEMS*, 189, FEB 15 2020.

[41] Yunhe Cui, Qing Qian, Chun Guo, Guowei Shen, Youliang Tian, Huanlai Xing, and Lianshan Yan. Towards ddos detection mechanisms in software-defined networking. *JOURNAL OF NETWORK AND COMPUTER APPLICATIONS*, 190, SEP 15 2021.

[42] Chunyang Fan, Jie Cui, Hulin Jin, Hong Zhong, Irina Bolodurina, and Debiao He. Auto-updating intrusion detection system for vehicular network: A deep learning approach based on cloud-edge-vehicle collaboration. *IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY*, 73(10):15372–15384, OCT 2024.

[43] Omar Elnakib, Eman Shaaban, Mohamed Mahmoud, and Karim Emara. Eidm: deep learning model for iot intrusion detection systems. *JOURNAL OF SUPERCOMPUTING*, 79(12):13241–13261, AUG 2023.

[44] Jagdeep Singh and Sunny Behal. Detection and mitigation of ddos attacks in sdn: A comprehensive review, research challenges and future directions. *COMPUTER SCIENCE REVIEW*, 37, AUG 2020.

[45] Ahmad Zainudin, Love Allen Chijioke Ahakonye, Rubina Akter, Dong-Seong Kim, and Jae-Min Lee. An efficient hybrid-dnn for ddos detection and classification in software-defined iiot networks. *IEEE INTERNET OF THINGS JOURNAL*, 10(10):8491–8504, MAY 15 2023.

[46] Usham Sanjota Chanu, Khundrakpam Johnson Singh, and Yambem Jina Chanu. A dynamic feature selection technique to detect ddos attack. *JOURNAL OF INFORMATION SECURITY AND APPLICATIONS*, 74, MAY 2023.

[47] Lotfi Mhamdi and Mohd Mat Isa. Securing sdn: Hybrid autoencoder-random forest for intrusion detection and attack mitigation. *JOURNAL OF NETWORK AND COMPUTER APPLICATIONS*, 225, MAY 2024.

[48] Jiangang Shu, Lei Zhou, Weizhe Zhang, Xiaojiang Du, and Mohsen Guizani. Collaborative intrusion detection for vanets: A deep learning-based distributed sdn approach. *IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS*, 22(7):4519–4530, JUL 2021.

[49] Xiaoyan Hu, Wenjie Gao, Guang Cheng, Ruidong Li, Yuyang Zhou, and Hua Wu. Toward early and accurate network intrusion detection using graph embedding. *IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY*, 18:5817–5831, 2023.

[50] Ankit Sharma and Manjeet Singh. Batch reinforcement learning approach using recursive feature elimination for network intrusion detection. *ENGINEERING APPLICATIONS OF ARTIFICIAL INTELLIGENCE*, 136(B), OCT 2024.

[51] Asma Alotaibi and Ahmed Barnawi. Securing massive iot in 6g: Recent solutions, architectures, future directions. *INTERNET OF THINGS*, 22, JUL 2023.

[52] Jiacheng Ye, Junshi Lv, Gaoming Xu, and Taijun Liu. Leaky cable perimeter intrusion detection based on deep reinforcement learning. *IEEE INTERNET OF THINGS JOURNAL*, 11(12):22616–22627, JUN 15 2024.

[53] Zhengfa Li, Chuanhe Huang, Shuhua Deng, Wanyu Qiu, and Xieping Gao. A soft actor-critic reinforcement learning algorithm for network intrusion detection. *COMPUTERS & SECURITY*, 135, DEC 2023.

[54] Jesus F. M. Cevallos, Alessandra Rizzardi, Sabrina Sicari, and Alberto Coen Porisini. Deep reinforcement learning for intrusion detection in internet of things: Best practices, lessons learnt, and open challenges. *COMPUTER NETWORKS*, 236, NOV 2023.

[55] Shoujian Yu, Xuan Wang, Yizhou Shen, Guowen Wu, Shui Yu, and Shigen Shen. Novel intrusion detection strategies with optimal hyper parameters for industrial internet of things based on stochastic games and double deep q-networks. *IEEE INTERNET OF THINGS JOURNAL*, 11(17):29132–29145, SEPT 1 2024.

[56] Chang Liu, Ruslan Antypenko, Iryna Sushko, and Oksana Zakharchenko. Intrusion detection system after data augmentation schemes based on the vae and cvae. *IEEE TRANSACTIONS ON RELIABILITY*, 71(2):1000–1010, JUN 2022.

[57] Lixiang Yuan, Siyang Yu, Zhibang Yang, Mingxing Duan, and Kenli Li. A data balancing approach based on generative adversarial network. *FUTURE GENERATION COMPUTER SYSTEMS-THE INTERNATIONAL JOURNAL OF ESCIENCE*, 141:768–776, APR 2023.

[58] He Zhang, Vishwanath Sindagi, and Vishal M. Patel. Image de-raining using a conditional generative adversarial network. *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY*, 30(11):3943–3956, NOV 2020.

[59] Ezzeldin Shereen, Kiarash Kazari, and Gyorgy Dan. A reinforcement learning approach to undetectable attacks against automatic generation control. *IEEE TRANSACTIONS ON SMART GRID*, 15(1):959–972, JAN 2024.

[60] Jie Ma and Wei Su. Collaborative ddos defense for sdn-based aiot with autoencoder-enhanced federated learning. *INFORMATION FUSION*, 117, MAY 2025.

[61] Samia Saidane, Francesco Telch, Kussai Shahin, and Fabrizio Granelli. Deep graphsage enhancements for intrusion detection: Analyzing attention mechanisms and gcn integration. *JOURNAL OF INFORMATION SECURITY AND APPLICATIONS*, 90, MAY 2025.

[62] Zixuan Wang, Zeyi Li, Mengyi Fu, Yingchun Ye, and Pan Wang. Network traffic classification based on federated semi-supervised learning. *JOURNAL OF SYSTEMS ARCHITECTURE*, 149, APR 2024.

[63] Jie Ma, Wei Su, Yikun Li, Yuan Yuan, and Ziqing Zhang. Synchronizing real-time and high-precision ldos defense of learning model-based in aiot with programmable data plane, sdn. *JOURNAL OF NETWORK AND COMPUTER APPLICATIONS*, 229, SEP 2024.

[64] Ahmad Zainudin, Rubina Akter, Dong-Seong Kim, and Jae-Min Lee. Federated learning inspired low-complexity intrusion detection and classification technique for sdn-based industrial cps. *IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT*, 20(3):2442–2459, SEP 2023.

[65] Zhimin He, Jie Yin, Yu Wang, Guan Gui, Bamidele Adebisi, Tomoaki Ohtsuki, Haris Gacanin, and Hikmet Sari. Edge device identification

based on federated learning and network traffic feature engineering. *IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING*, 8(4):1898–1909, DEC 2022.

[66] Othmane Friha, Mohamed Amine Ferrag, Lei Shu, Leandros Maglaras, Kim-Kwang Raymond Choo, and Mehdi Nafaa. Felids: Federated learning-based intrusion detection system for internet of things. *JOURNAL OF PARALLEL AND DISTRIBUTED COMPUTING*, 165:17–31, JUL 2022.

[67] Pan Wang, Zixuan Wang, Feng Ye, and Xuejiao Chen. Bytesgan: A semi-supervised generative adversarial network for encrypted traffic classification in sdn edge gateway. *COMPUTER NETWORKS*, 200, DEC 9 2021.

Chen Jue received the Ph.D. degree in Computer Science from East China Normal University, Shanghai, China, in 2018. He is currently a lecturer with the Shanghai University of Engineering Science, Shanghai, China.He has published multiple papers in referred journals and conferences, including Computer Communications (JCR region 1 journal), etc. Dr. Chen is an Active Reviewer of several international journals, including ACM Transactions on Cyber-Physical Systems, etc. His research interests include Software Defined Network (SDN) and Artificial Intelligence.

Tao Hongyu received his B.S.degree in Shanghai University of Engineering Science, Shanghai, China, in 2022. He is currently pursuing the M.S. degree in Electronic Information from Shanghai University of Engineering Science, Shanghai, China, since 2023. His main research interests is Software Defined Network and Intrusion detection.

Cui Meng received his M.S.degree in Control Science and Engineering from Shanghai University of Engineering Science, Shanghai, China, in 2024. His main research interests include Software Defined Networks and intrusion detection.

Peng Haidong received his B.S.degree in Shanghai University of Engineering Science, Shanghai, China, in 2023. He is currently pursuing the M.S. degree in Electronic Information from Shanghai University of Engineering Science, Shanghai, China, since 2024. His main research interests is Software Defined Network, Artificial Intelligence and Intrusion detection.

Qiu Xihe received her Ph.D. degree from National University of Singapore, Singapore, in 2018. Currently, she is an Associate Professor with the School of Electronic and Electrical Engineering, Shanghai University of Engineering Science, Shanghai, China. Her research mainly focuses on intelligent clinical decision support and deep reinforcement learning.