Peer-to-Peer Networking and Applications
https://doi.org/10.1007/s12083-023-01595-6

Pre-trained language model-enhanced conditional generative adversarial networks for intrusion detection

Fang Li[1] · Hang Shen[1] · Jieai Mai[1] · Tianjing Wang[1] · Yuanfei Dai[1] · Xiaodong Miao[2]

Received: 10 August 2023 / Accepted: 14 November 2023

Abstract
As cyber threats continue to evolve, ensuring network security has become increasingly critical. Deep learning-based intrusion detection systems (IDS) are crucial for addressing this issue. However, imbalanced training data and limited feature extraction weaken classification performance for intrusion detection. This paper presents a conditional generative adversarial network (CGAN) enhanced by Bidirectional Encoder Representations from Transformers (BERT), a pre-trained language model, for multi-class intrusion detection. This approach augments minority attack data through CGAN to mitigate class imbalance. BERT with robust feature extraction is embedded into the CGAN discriminator to enhance input–output dependency and improve detection through adversarial training. Experiments show the proposed model outperforms baselines on CSE-CIC-IDS2018, NF-ToN-IoT-V2, and NF-UNSW-NB15-v2 datasets, achieving F1-scores of 98.230%, 98.799%, and 89.007%, respectively, and improving F1-scores over baselines by 1.218%−13.844% 0.215%−13.779%, and 2.056%−22.587%.

Keywords Intrusion detection · Multi-class classification · Bidirectional encoder representations from transformers (BERT) · Conditional generative adversarial network (CGAN)

1 Introduction

Ensuring network security is becoming more critical with the continuous evolution of cyber threats. The 2022 Cyberthreat Defense Report[1] released by CyberEdge organization shows that successful cyberattacks remain prevalent, with over 85% of organizations experiencing at least one attack in the past year, and 40.7% have dealt with six or more attacks. The top concerns are ransomware, malware, and account takeover attacks, which can severely affect national security, economic development, and social stability. Therefore, effectively detecting intrusions in computer networks has become an urgent challenge.

Intrusion detection systems (IDS) aim to identify malicious network activities by monitoring traffic patterns and system operations. They generate alerts when attacks are detected, allowing security personnel to take timely response [1]. Machine learning, especially deep learning, has emerged as an effective approach for building IDS models by automatically learning representations from network traffic data [2–4]. Unlike traditional shallow learning models, deep

This article is part of the Topical Collection: *Special Issue on 2 - Track on Security and Privacy*
Guess Editor: Rongxing Lu

✉ Hang Shen
hshen@njtech.edu.cn

Fang Li
202261220027@njtech.edu.cn

Jieai Mai
maijieai@outlook.com

Tianjing Wang
wangtianjing@njtech.edu.cn

Yuanfei Dai
daiyuanfei@njtech.edu.cn

Xiaodong Miao
mxiaodong@njtech.edu.cn

[1] College of Computer and Information Engineering (College of Artificial Intelligence), Nanjing Tech University, Nanjing 211816, China

[2] School of Mechanical and Power Engineering, Nanjing Tech University, Nanjing 211816, China

[1] https://cyber-edge.com/wp-content/uploads/2022/11/CyberEdge-2022-CDR-Report.pdf

Published online: 25 November 2023

neural networks can capture complex non-linear dependencies and extract informative features through multiple layers tailored for high-dimensional learning. This alleviates the need for manual feature engineering, enabling end-to-end learning of IDS models directly from traffic data.

However, building highly accurate IDS remains challenging due to the increasing complexity and diversity of attacks. Conventional machine learning approaches (e.g., support vector machine [5] and naive Bayes [6]) often fail to provide satisfactory performance when dealing with massive, high-dimensional network data with complex features. Though deep learning methods can learn inherent representations from traffic data through nonlinear structures to meet the demands of high-dimensional learning and prediction, the continuous evolution of new attack types poses significant challenges for intrusion detection based on deep learning.

1.1 Motivations and related works

Building highly accurate IDS remains challenging due to the increasing complexity and diversity of attacks, in which imbalanced data and insufficient feature extraction ability are important factors that limit performance.

1.1.1 Network traffic generation

Network intrusion data often exhibits class imbalance, where attacks only account for a small fraction compared to normal traffic. Moreover, different attack types vary greatly in sample size and intrinsic patterns. Relying heavily on samples, deep learning models trained on such imbalanced datasets tend to underfit minority attack classes, resulting in insufficient feature learning and low detection accuracy.

A common solution is to balance the data through oversampling and undersampling techniques. The Synthetic Minority Oversampling TEchnique (SMOTE) [7] was used to address the problem of network traffic data imbalance [8]. Wu et al. [9] proposed a network intrusion detection algorithm based on the enhanced random forest and SMOTE, which used a hybrid algorithm combining the k-means clustering algorithm with SMOTE sampling to increase the number of minor samples and thus achieved a balanced dataset. In [10], training an ensemble classifier with undersampling data and each sub-ensemble resolved the issue of minority attack classes. Oversampling aims to duplicate samples from the minority class. In contrast, undersampling seeks to remove samples from the majority class. The former is prone to overfitting, whereas the latter reduces samples.

Another line of research is to generate more attack samples of minority classes using the generative adversarial network (GAN) [11, 12]. The samples were merged into the original dataset, and the combined dataset was used to train a multi-classification model for various attacks. For the robustness of detection systems, IDSGAN [13] was proposed to generate adversarial malicious traffic records aiming to attack intrusion detection systems by deceiving and evading the detection. In [14], the author proposed a tabular data sampling method to balance normal and attack samples. A k-nearest neighbor method was used for effective undersampling for normal samples, while a tabular auxiliary classifier GAN model was designed for attack sample oversampling. He et al. [15] proposed a conditional GAN (CGAN)-based collaborative intrusion detection algorithm with blockchain-empowered distributed federated learning, which introduces long short-term memory (LSTM) into the CGAN training to improve the effect of generative networks and the generated data are used as augmented data and applied in the detection and classification of intrusion data.

1.1.2 Deep features extraction

Informative features must be extracted to clarify class boundaries in the low-dimensional space. One solution is to map the data into a high-dimensional space through feature extraction, making the boundaries among different attacks more separable. Deep neural networks can learn to extract complex multi-dimensional representations from traffic data, transforming attack types into distinct locations in a high-dimensional space.

Typical techniques include convolutional neural networks (CNN) and LSTM [16]. The former extracts features of multi-dimensional tensors through convolution. The latter can deeply mine data's temporal and semantic information. Steven et al. [17] proposed a character-level IDS based on CNN, which treats network traffic records as character sequences. Character sequences were encoded as alphabet-based vectors and aggregated into a matrix for input to CNN for classification. Aydın et al. [18] designed an LSTM-based system for the detection and prevention of DDoS attacks in a public cloud network environment and developed an LSTM prediction model for the system. Bi-directional LSTM (BiLSTM) reinforces the attention on feature backward dependency [19]. Some researchers attempted to solve the binary and multi-classification problem of network traffic with LSTM and BiLSTM [20–23]. Imrana et al. [23] proposed to use a BiLSTM model for network intrusion detection, which achieved better accuracy than conventional LSTM. A weighted-intrusion-based cuckoo search with a graded neural network is presented in [24] to identify and categorize the anomalies in a supervisory control and data acquisition system through data optimization.

However, due to data complexity and attack diversity, boosting feature extraction ability still faces many challenges. Large language models [25] possess robust semantic understanding and generation abilities in natural language processing (NLP). Researchers adopted universal architectures like

Transformer, not relying on domain-specific knowledge, and thus can generalize to various domains. Bidirectional Encoder Representations from Transformers (BERT) [26] is a pre-trained language representation model originally developed by Google for natural language processing (NLP) tasks. Compared to BiLSTM and LSTM, BERT adopts the Transformer architecture with self-attention to capture long-range dependencies better. With more parameters, BERT can learn semantic representations more effectively and achieve state-of-the-art results on various NLP tasks including text classification and prediction. Given its robust feature extraction and generalization capabilities, researchers have recently explored applying BERT to other domains, such as vulnerability prediction and log anomaly detection. Jiao et al. [27] proposed ExBERT, which uses the collected vulnerability description corpus to fine-tune the pre-trained BERT model, aiming to extract the semantic information of vulnerability descriptions to predict network security vulnerabilities. LAnoBERT [28] is a BERT-based log anomaly detection model that detects log parsing free, where BERT extracts semantic information in serialized log data and captures detailed features to improve the accuracy of log anomaly detection. Alkhatib et al. [29] leveraged a BERT-based architecture to detect message injection intrusions in the controller area network bus. However, few studies consider effectively integrating pre-trained language models, such as BERT, with existing deep learning models or frameworks.

1.2 Contributions and organization

Unlike existing research solutions, our proposal employs a pre-trained language model in NLP to enhance feature extraction and data generation. The purpose is to boost multi-class intrusion detection through pre-trained language model-assisted adversarial learning. The main contributions of this work are three-fold:

- We present a BERT-enhanced CGAN framework that leverages CGAN to augment minority attack samples for balancing training data and integrates BERT's powerful feature representations to boost detection accuracy.
- The BERT model is embedded in the CGAN discriminator to extract informative features from generated samples, which improves the CGAN generator to produce more realistic samples through adversarial training.
- Experimental results on multiple authoritative datasets demonstrate that the proposed scheme is superior to baseline approaches in accuracy, precision, recall, and F1-score. It improves the weighted F1-score for multi-class classification on CSE-CIC-IDS2018[2] by 1.218%−13.844%. On NF-ToN-IoT-v2,[3] and NF-UNSW-NB15-v2,[4] the improvement ranges are 0.215%−13.779% and 2.056%−22.587%.

The follow-up structure of this paper is organized as follows. Section 2 introduces the preliminaries for CGAN and BERT models. The proposed framework is discussed in Section 3. Section 4 presents the dataset selection and experimental methodology. We conduct experimental results and performance evaluation in Section 5, followed by the concluding remarks in Section 6.

2 Preliminaries

2.1 Conditional GAN (CGAN)

A GAN consists of a generator (Generator, G) and a discriminator (Discriminator, D), both of which are trained in an adversarial game [11]. With the ability to augment samples, the GAN has become a widely used technique for expanding datasets. The GAN can generate samples close to the actual probability distribution by training with actual data. The augmented dataset makes the trained model more generalizable. The GAN framework can be formulated as a two-player min-max game between a discriminative network, $D(x)$, and a generative network, $G(z)$, given by

$$\min_G \max_D V(D,G) = E_{x\sim p(x)}\left[\log D(x)\right] + E_{z\sim p(z)}\left[\log\left(1 - D(G(z))\right)\right]. \tag{1}$$

In the discriminator, x is input from the accurate distribution, $p(x)$, to a discriminative function. z denotes the input noise. The generator's $p(z)$ represents prior noise distribution.

The CGAN model [30] extends GAN with conditional control, allowing us to condition the network with additional information for generating different types of network traffic. Some investigators studied CGAN-based dataset expansion [31]. The input of G includes random noise and additional conditional information, y, which can be any auxiliary information (e.g., class labels or data from other modalities). Unlike GAN, whose input is only random noise, CGAN can feed y as an additional input layer to the discriminator and the generator. The optimization objective of both G and D for CGAN can be expressed as

$$\min_G \max_D V(D,G) = E_{x\sim p(x)}\left[\log D(x|\mathrm{y})\right] + E_{z\sim p(z)}\left[\log\left(1 - D(G(z|\mathrm{y}))\right)\right]. \tag{2}$$

[2] https://www.unb.ca/cic/datasets/ids-2018.html

[3] https://rdm.uq.edu.au/files/a4ad7080-ef9c-11ed-a964-b70596e96ad5

[4] https://rdm.uq.edu.au/files/8c6e2a00-ef9c-11ed-827d-e762de186848

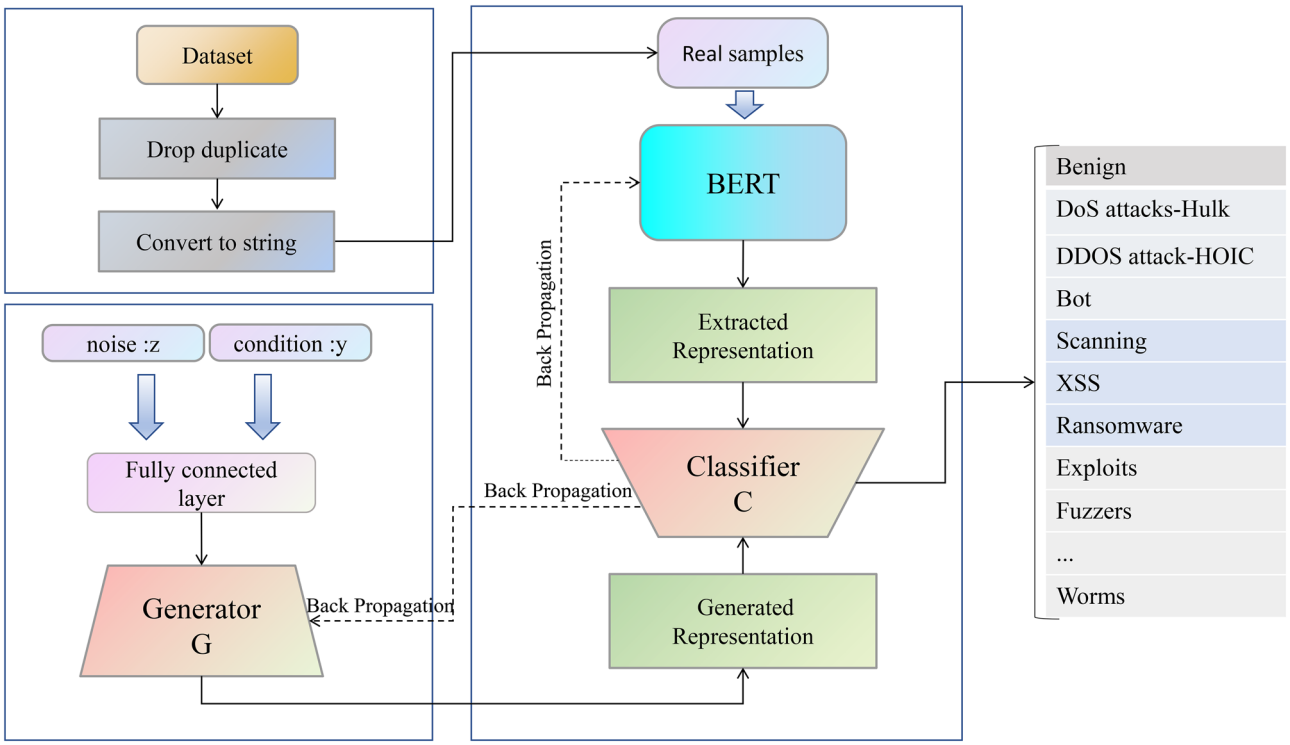


Fig. 1 Pre-trained language model-enhanced adversarial training framework for intrusion detection

2.2 Transformer and BERT

A Transformer [32] is a deep learning model that adopts a mechanism of self-attention-based encoder-decoder. This model relies entirely on the attention mechanism to map the global dependencies between input and output. The encoder maps the input sequence of symbols into a continuous high-dimensional representation. Given $\{x_1, ..., x_n\}$, the decoder generates an output sequence, $z = \{z_1, ..., z_n\}$. The encoder and decoder support stacked self-attention and point-wise. The Transformer combines two attention mechanisms, Scaled Dot-Product Attention and Multi-Head Attention. The former queries all keys by dot product and calculates the weight value via the softmax function. The latter allows the model to jointly attend to information from different representation subspaces at various locations. The self-attention layer connects all the neural network structure layer positions with several sequentially executed operations. This connection makes the feature information learned by different neural network layers before and after be paid attention by the self-attention layer, which is conducive to establishing long-range dependencies between input and output sequences. The self-attention mechanism enables the Transformer to extract more hidden features for learning.

BERT inherits the Transformer's self-attention-based architecture and implements pre-trained deep bidirectional representations. It has become a popular deep neural network model in NLP. With powerful feature extraction, BERT can learn more about the global input and output dependencies. The fine-tuned BERT can be used to handle specific tasks. The pre-trained model parameters are used to initialize the BERT model. After initialization, labeled data from downstream tasks are used to fine-tune the parameters of BERT end-to-end.

3 Solution

Generally, generating a specific proportion of each attack type through GAN requires separate training [12]. Conditional GANs (CGANs) allow controlling the categories and proportions of generated anomalous attacks, supporting diverse classification. However, accurately identifying attacks with inconspicuous features remains difficult. To address this, we embed BERT into the CGAN discriminator to enhance intrusion detection. BERT can capture informative features from network traffic, alleviating performance degradation due to class imbalance.

As illustrated in Fig. 1, the proposed intrusion detection framework includes three parts:

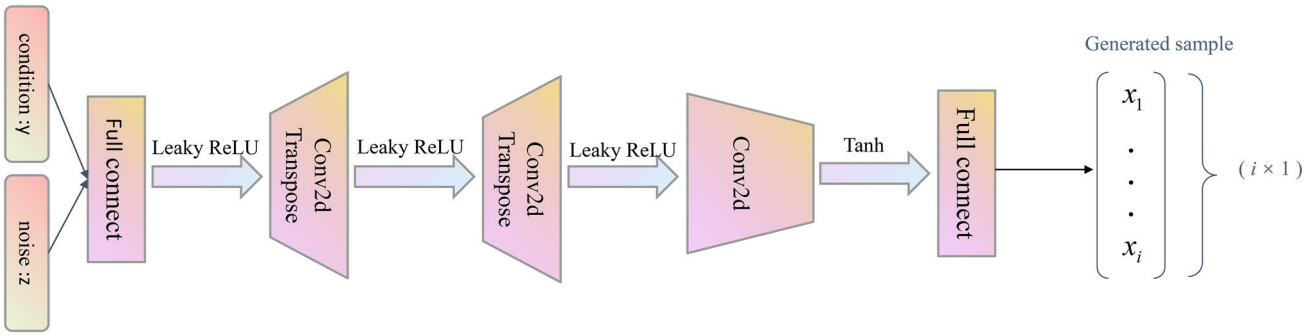


Fig. 2 Generator network structure

- **Data preprocessing** converts non-textual network traffic statistical feature data into text-formatted data to adapt to the large-scale language model BERT (see Section 3.1).
- **Generator** generates different kinds of network attack traffic samples based on conditional control information to augment the dataset (to be discussed in Section 3.2).
- **BERT-enhanced classifier** extracts network traffic features and encodes them into high-dimensional representations. The module classifies the high-dimensional feature representations from BERT and the generator (see Section 3.3).

3.1 Data preprocessing

The chosen datasets have ten types of network traffic, including benign data and nine abnormal attacks, in which duplicate data is removed. Acceptable input types for BERT are strings. Features not in character form are converted to characters that BERT can handle. Each piece of traffic data after conversion corresponds to a sentence, and the statistical features of the traffic data correspond to the words in the sentence. BERT extracts hidden features from sentences, equivalent to pulling high-dimensional feature representations from network traffic data. Finally, the labels have been encoded as one-hot vectors to support classification training.

The proportion of benign data flow in real-world networks exceeds that of abnormal attack data flow. When processing the data, we do not excessively reduce the benign traffic data and maintain the normal state of benign and redundant anomalous attack data.

3.2 Generator

In the CGAN framework, network traffic class labels are selected as conditional control information y, input to the generator. The other input to the generator, G, comes from a random noise vector in the prior space $p(z)$. The generator uses the input to generate a new high-dimensional feature representation, expressed by $g = G(z, y)$.

The generator G and discriminator D are trained alternately, and the discriminator is optimally trained before the generator parameter update. In this case, the generator must minimize the Jensen-Shannon (JS) dispersion between the real and the generated traffic, expressed as

$$\min_G V(D, G) = -\log 4 + 2 \cdot \mathrm{JSD}(p_x || p_z) \tag{3}$$

where $\mathrm{JSD}(p_x || p_z)$ calculates the similarity degree between real traffic x and noise z of the probability density distributions.

Figure 2 shows the structure diagram of the generator. The noise vector z and the network attack class label information y are inputted into the network together. The first fully connected layer uses the Leaky ReLU activation function. When a negative value occurs in the network parameter update process, the gradient of the Leaky ReLU activation function still exists, avoiding that the gradient of the ordinary ReLU activation function is zero and the parameters cannot be updated when the input is negative. The Reshape layer converts the shape to 2D. The two-dimensional matrix is upsampled by deconvolution, the convolution kernel size is 4×4, the stride is 2×2, and the activation function is Leaky ReLU. To ensure the diversity of the generated data, deconvolution upsampling is repeated, the convolution kernel size is 4×4, the stride is 2×2, and the activation function is Leaky ReLU. After two deconvolution upsampling, convolution is used for downsampling. The convolution kernel size is 5×5, and the stride is 1×1, with activation function tanh. Finally, through the fully connected layer, the output result is obtained.

3.3 BERT-enhanced classifier

The discriminator network of most GANs only supports binary classification that outputs whether the traffic is real or generated. We consider a discriminator that supports

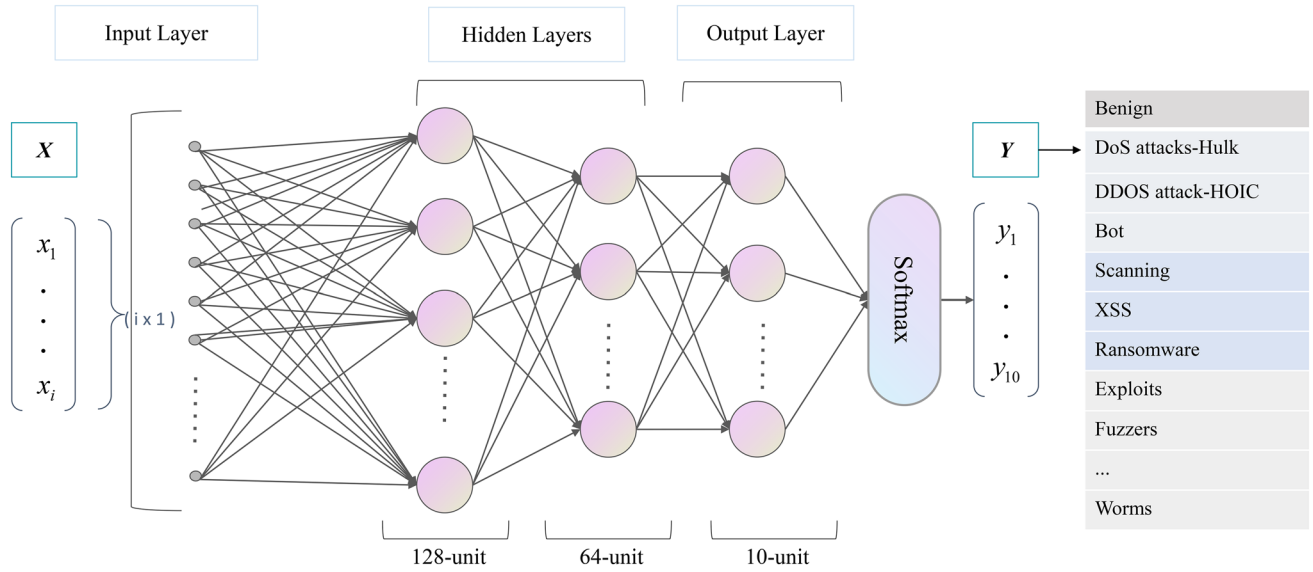


Fig. 3 Classification network structure

multi-class classification (a type of classifier, C), which uses three fully connected layers, as shown in Fig. 3, to predict whether the data is normal data or a specific type of network attack. The last fully connected layer has ten output units, and the activation function uses the Softmax function.

Instead of being used for NLP, BERT, in this work, is embedded into the CGAN classification network layer for solving network traffic detection. For a classification network layer with BERT embedded, the output of the network layer needs to be adapted to support network traffic classification tasks. As shown in Fig. 4, BERT and C are combined into detection modules to classify network traffic data. The first token of every sequence is always a special classification token ([CLS]). The final hidden state for the token is utilized as the aggregate sequence representation for classification tasks. Sequences are separated with a special token ([SEP]). The parameters of the pre-trained model are used to initialize BERT. Then, the preprocessed network intrusion dataset is used to fine-tune BERT. Compared with randomly initializing BERT parameters, using the parameters of a pre-trained model can accelerate the learning of traffic features, conducive to fast convergence.

Real traffic data, x, is considered a sentence token to be processed in the BERT model. Special characters, such as [CLS] and [SEP], are added before and after each traffic data sequence to facilitate BERT's recognition. Three matrices are used in multi-head attention to calculate the attention score among traffic data sequences. The embedded sequence of x is multiplied by the weight matrices $\mathbf{W}_Q \in R^{d\times d_q}$, $\mathbf{W}_K \in R^{d\times d_k}$ and $\mathbf{W}_V \in R^{d\times d_v}$, where $\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$ are the query, key and value matrix respectively, and the $\mathbf{W}_Q$, $\mathbf{W}_K$ and $\mathbf{W}_V$ are their trainable weight matrices. For each header, the self-attention function is performed by inputting the embedded sequence of x to get a new vector, given by

$$\mathbf{head}_i = softmax(\frac{\mathbf{Q}\cdot\mathbf{K}^T}{\sqrt{d_k}})\mathbf{V}. \tag{4}$$

The vector at the [CLS] token position of the hidden state at the last layer is used as the semantic representation of traffic data. The classifier handles h and g. h is the probability distribution $p(x)$ of the preprocessed traffic data mapped to a high-dimensional space by BERT encoding. $g = G(z, y) \in R^d$ is the high-dimensional feature representation generated by the generator according to class label y and random noise vector z. The probability of classifier output $Y = C(\mathrm{x})$ represents the traffic type. During training, the generator attempts to generate high-dimensional feature representations of network traffic to confuse the classifier, which class of traffic data the classifier tries to distinguish correctly.

The detection model composed of BERT and classifier C is trained end-to-end using cross-entropy loss function, and the Adaptive Moment Estimation (Adam) algorithm as in [33] is used for parameter update. The loss function is expressed as

$$L_{\mathrm{c}} = -E_{x\sim p(x)}[\log C(x)]. \tag{5}$$

C needs to distinguish the categories of feature representations that BERT encodes the network traffic data to form a high-dimensional space. At the same time, the classifier needs to distinguish the categories of the samples generated by G. The objective functions of G and C for the min-max optimization are formulated as

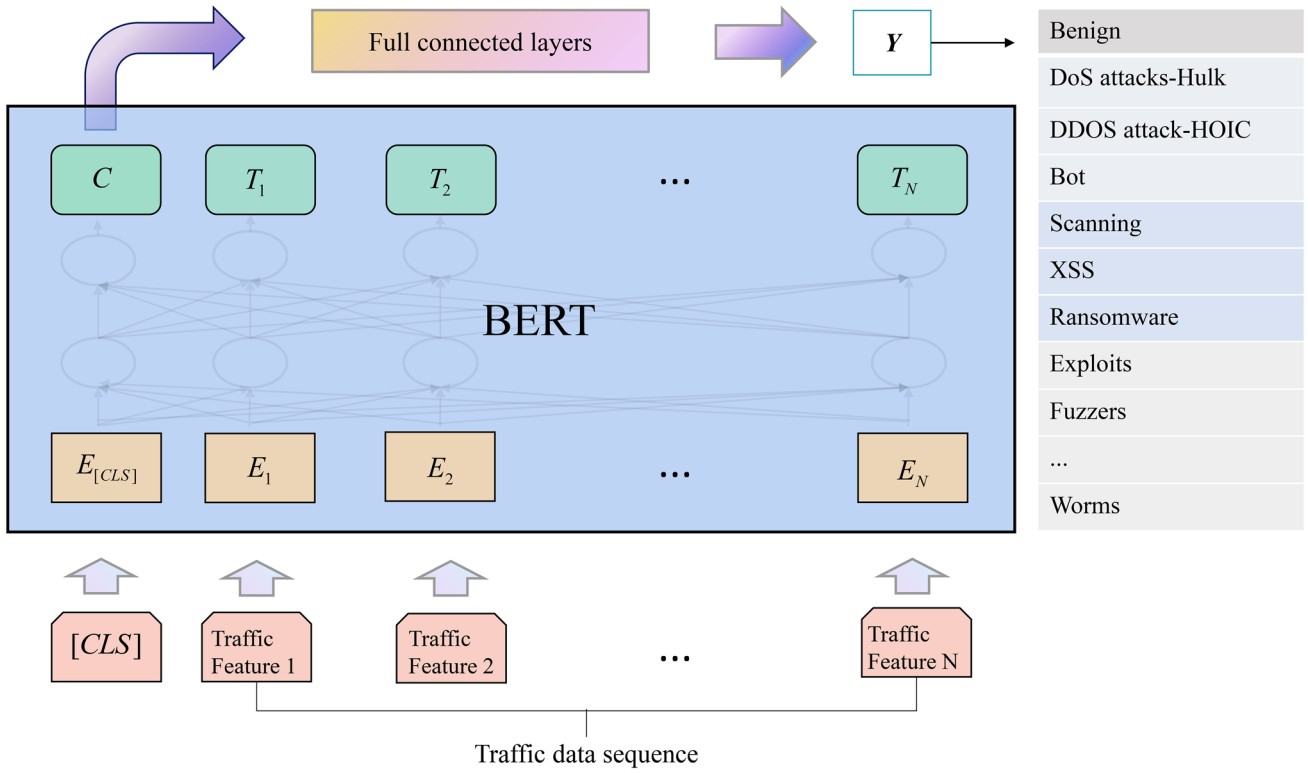


Fig. 4 BERT for supporting network traffic classification: Each sequence corresponds to a piece of network traffic data. Features represent statistical characteristics of network traffic

$$\min_G \max_C V(C,G) = E_{x\sim p(x)}\left[\log C(x)\right] + E_{z\sim p(z)}\left[\log\left(1 - C(G(z,\mathrm{y}))\right)\right]. \tag{6}$$

When training is complete, the classifier can distinguish between benign network traffic and different types of network attacks. At the same time, the high-dimensional feature representation of network traffic generated by G becomes approximately the proper probability distribution, $p(x)$.

In the BERT-enhanced classifier, the role of BERT is not simply used to enhance the feature extraction ability for network traffic. More importantly, BERT forms a mutually reinforcing relationship with G. Specifically, to learn the proper probability distribution traffic features, the generator generates a high-dimensional feature representation of network traffic that changes along with the fine-tuning of BERT parameters. It continuously learns the high-dimensional feature representation of network traffic. The classifier also constantly updates the parameters.

3.4 Training strategy

The CGAN generator must learn the high-dimensional feature representation of network traffic known by BERT. During training, the GAN freezes the parameters of one part (G or C) and trains the parameters of the other. The two sides alternate until the training is complete. CGAN, derived from GAN, follows the training policy. In our scheme, BERT is embedded in CGAN, and the entire model training is still to update the parameters alternately between G and C. The implementation details of the proposed method fusion training algorithm are summarized in Algorithm 1. First, the BERT model is initialized with pre-trained model parameters, and the parameters of G and C are randomly initialized. Since BERT did not learn useful information before fine-tuning, the first step is to fine-tune BERT and update the C's parameters, see lines 3-7. After the first step, BERT learns the high-dimensional feature representation of network traffic and then enters the second step to train CGAN, see lines 9-16. The two phases are alternated until the training of the proposed model is completed.

Training a GAN model and a classification network are separated in some existing solutions [12, 34]. These methods use the trained GAN to generate samples and then prepare a classification network using the augmented dataset. Due to the separate training processes, the number of generated samples can be set as required.

Input: $< \mathrm{x}_i, \mathrm{y}_i >, < z_i, c_i > (i = 1, 2, ..., m)$, where m denotes the size of each batch, c_i denotes the conditional information.
Output: $\{w, p, q\}$, where w, p, and q denote parameters of BERT, C, and G.
Init: BERT with pre-training model parameters, G and C with stochastic initialization.
for $i = 1...I$ **do**
  for $j = 1...J_1$ **do**
    $h_i \leftarrow \mathrm{BERT}(x_i)$;
    $\hat{\mathrm{y}}_i \leftarrow C(h_i)$;
    Calculate $L(\mathrm{y}_i, \hat{\mathrm{y}}_i)$ based on (5);
    Update w and p via Adam;
  end for
  for $j = 1...J_2$ **do**
    $h_i \leftarrow \mathrm{BERT}(x_i)$;
    $g_i \leftarrow G(z_i, c_i)$;
    $m_i \leftarrow h_i \cup g_i$;
    $\hat{\mathrm{y}}_i \leftarrow C(m_i)$;
    Calculate $L(\mathrm{y}_i, \hat{\mathrm{y}}_i)$ based on (5);
    Freeze q and update p via Adam;
    Freeze p and update q via Adam;
  end for
end for

Algorithm 1 Fusion training

Unlike existing frameworks, the traffic type prediction and sample generation (training a CGAN and a classification network) in the proposed framework are fused. The CGAN model continuously updates parameters during training to generate samples closer to the actual probability distribution. At the same time, the classification network classifies data from BERT and G. The initially generated samples are immature and different from the actual probability distribution. With the update of model parameters, the generated samples become realistic. Accordingly, the quantity and quality of generated samples are gradually determined during training rather than being set by the experimenter. Through the fusion training, the classification network is promoted to classify traffic types efficiently.

For the model training, it needs to compute and update the gradients of the BERT, C, and G. The number of floating-point operations to update the BERT, C, and G is $O(w + p + q)$. When batch size is set to m, each epoch generates $m \cdot O(w + p + q)$ floating-point operations. Accordingly, the total computation complexity of the proposed method is $O((m \cdot (w + p + q) \cdot N_e)$, where N_e is the total number of training epochs.

Table 1 Data distribution of three datasets

Dataset	Network traffic type	Training set	Test set
CSE-CIC-IDS2018	Benign	360162	183683
	DoS attacks-Hulk	80391	40187
	DDOS attack-HOIC	61670	30828
	DDoS attacks-LOIC-HTTP	43214	21607
	SSH-Bruteforce	40314	20159
	Infilteration	36275	18109
	Bot	35743	17868
	FTP-BruteForce	28009	14112
	DoS attacks-GoldenEye	16598	8302
	DoS attacks-SlowHTTPTest	13416	6731
	In total	715792	361586
NF-ToN-IoT-v2	Benigh	129636	32409
	Scanning	80352	20088
	XSS	52164	13041
	DDoS	43056	10764
	Password	24516	6129
	DoS	15156	3789
	Injection	14544	3636
	Backdoor	360	90
	MITM	144	36
	Ransomware	72	18
	In total	360000	90000
NF-UNSW-NB15-v2	Benign	120000	37000
	Exploits	20509	11042
	Fuzzers	14505	7805
	Generic	10770	5790
	Reconnaissance	8337	4442
	DoS	3773	2021
	Analysis	1446	853
	Backdoor	1416	753
	Shellcode	937	490
	Worms	91	73
	In total	181784	70269

4 Experimental preparation

4.1 Dataset selection

We evaluated the detection performance on three challenging datasets: CSE-CIC-IDS2018, NF-ToN-IoT-v2, and NF-UNSW-NB15-v2. Table 1 lists the data distribution in each dataset, with different degrees of data imbalance. Most of the CSE-CIC-IDS2018 and NF-UNSW-NB15-v2 datasets belong to normal traffic, especially the latter, where the

proportion of normal traffic is as high as 66.01%, and the total proportion of attack traffic is less than 35%. For the NF-ToN-IoT-v2 dataset, normal traffic accounts for 36.01%, the highest among the ten categories, and the least attack category accounted for only 0.02%.

4.2 Implementation details

The preprocessed datasets were used for experiments and performance evaluation. Due to category imbalance, if the data is randomly sampled during model training, those categories that occupy a minority may not be extracted within a batch. For this situation, we rewrote select_sample() to set the number and proportion of each attack category in each batch as needed.

The number of network data in a batch was fixed at 100 in the experiment. The data of each batch was obtained by random sampling. For CSE-CIC-IDS2018, there were 50 pieces of data in a batch for Benign and three for DoS attacks-SlowHTTPTest, a type of attack with minor data. The total number of data entries for Benign-type attacks in the training set is 360162. About 50 Benign type data were randomly selected for a batch. The rest of the categories followed the same approach.

In the proposed framework, BERT, as feature extraction network layers, combines network output units composed of fully connected layers to perform multi-classification of network traffic. The number of layers (i.e., Transformer blocks), the hidden size, and self-attention heads are denoted as L, H, and A, respectively. We primarily report the setting on model sizes: BERT ($L = 2$, $H = 256$, $A = 4$; The total number of parameters is about 10 M).

All models in our experiments ran on Python 3.5, Tensorflow 2.12.0 environment. The computer configuration is Intel Core i9-13900K CPU, RTX 4090 24 G, and 128GB RAM.

4.3 Baseline selection and parameter setting

For comprehensive comparison and verification, we selected and designed five baseline approaches in the following:

- LSTM, utilized to solve the binary and multi-classification of network traffic in [18]. The LSTM benchmark consists of four LSTM layers. The units of each layer are 50, 50, 30, and 30, and the total amount of parameters is 66K.
- BiLSTM, used to multi-class intrusion detection in [23]. This benchmark is also composed of four LSTM layers. Because it learns long-term bidirectional dependencies, the units of each layer are 50×2, 50×2, 30×2, and 30×2, respectively. The total amount of model parameters is 169K.
- LSTM-CGAN: LSTM can also serve as feature extractors. We embed LSTM into the CGAN discriminator and compare it with the proposed solution to evaluate the collaborative performance of CGAN and BERT.
- BiLSTM-CGAN: Like LSTM-CGAN, BiLSTM is embedded into the CGAN discriminator to compare the proposed solutions further.
- BERT, which is consistent with BERT embedded in the proposed framework.

4.4 Evaluation metrics

Indicators such as accuracy, precision, recall, and F1-score, widely adopted in related fields, are used for performance evaluation. Accuracy is the most intuitive evaluation index to reflect the model's performance. For unbalanced data classes, F1-score, precision, and recall complement each other to evaluate model performance comprehensively. The following are the accuracy, precision, recall, and F1-score calculation formulas.

- Accuracy: This is quantified as the ratio of the number of correct network traffic to the total number of classification predictions.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{7}$$

- Precision: This is the proportion of correct classifications out of a set of predictions classified as attacks.

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

- Recall: This is quantified as the proportion of correct attack classifications from a given set of attack instances.

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

- F1-score: This is defined as the harmonic mean of precision and recall, comprehensively reflecting the effect of the model from precision and recall.

$$\text{F1 - score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{10}$$

Denote TP, TN, FP, and FN as true positive, true negative, false positive, and false negative, respectively.

From (6), precision reflects positive predictive value. Higher precision means a lower probability of false positives. Recall emphasizes the actual positive rate. The higher the recall is the lower likelihood of false negatives. As a comprehensive indicator, F1-score reflects model precision

Table 2 Weighted average performance of different approaches on three datasets

Dataset	Method	Accuracy	Precision	Recall	F1-score
CSE-CIC-IDS2018	Proposed	98.218%	98.247%	98.218%	98.230%
	BERT	96.828%	97.463%	96.828%	97.012%
	BiLSTM-CGAN	88.665%	87.677%	88.665%	86.918%
	LSTM-CGAN	88.410%	87.077%	88.410%	86.579%
	BiLSTM	86.717%	85.911%	86.717%	85.200%
	LSTM	86.070%	86.107%	86.070%	84.386%
NF-ToN-IoT-v2	Proposed	98.797%	98.805%	98.797%	98.799%
	BERT	98.566%	98.628%	98.566%	98.584%
	BiLSTM-CGAN	87.026%	86.888%	87.026%	86.325%
	LSTM-CGAN	86.893%	87.001%	86.893%	86.180%
	BiLSTM	86.311%	86.709%	86.311%	85.582%
	LSTM	85.888%	86.219%	85.888%	85.020%
NF-UNSW-NB15-v2	Proposed	87.398%	91.688%	87.398%	89.007%
	BERT	86.064%	89.271%	86.064%	86.951%
	BiLSTM-CGAN	73.737%	76.181%	73.737%	72.631%
	LSTM-CGAN	69.381%	70.858%	69.381%	67.597%
	BiLSTM	71.599%	73.699%	71.599%	70.538%
	LSTM	68.386%	68.336%	68.386%	66.420%

and recall and is used to evaluate network intrusion detection experiments. The higher the F1-score, the more types of network attacks the model can correctly identify.

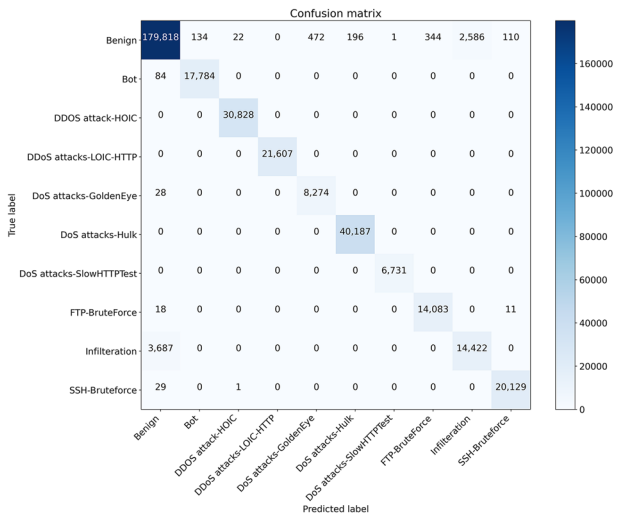
5 Experimental results

We evaluated the multi-class classification performance of different methods on three datasets by having the models predict whether a sample was normal traffic or one of the attack traffic types provided in the dataset (ten classes for CSE-CIC-IDS2018, NF-ToN-IoT-v2, and NF-UNSW-NB15-v2). This section first compared different methods' weighted average classification performance on the test set for all network traffic types. Then, the precision, recall, and F1-score of other ways of detecting specific types of network traffic were analyzed to evaluate the proposed scheme's detection performance against different attacks.

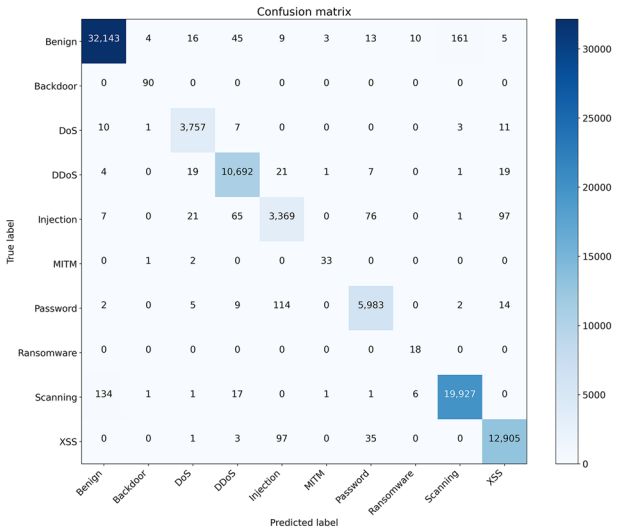
5.1 Overall performance analysis

Table 2 shows the accuracy, precision, recall, and F1-score of different methods on the test set of CSE-CIC-IDS2018, NF-ToN-IoT-v2, and NF-UNSW-NB15-v2. The proposed model outperforms other methods in all evaluation metrics, followed by BERT, BiLSTM, and LSTM at the lowest. Compared with BiLSTM-CGAN, the accuracy of the proposed method for those three datasets was improved by 9.553%, 11.771%, and 13.661%, respectively. This is due to the self-attention mechanism of BERT, which allows for a model of dependencies without regard to the distance of features in the input or output sequence [32]. BERT can capture the intrinsic connection among the network statistical features through the self-attention mechanism, even for the most distant network traffic statistical features in a sequence. The classifier integrated with BERT can obtain more information about the attack category from the captured high-dimensional features, improving classification precision. Compared with BERT, BiLSTM and LSTM have a weaker ability to pay attention to the global dependencies between network statistical features. When faced with a small proportion of attack categories, they need help distinguishing the attack types correctly.

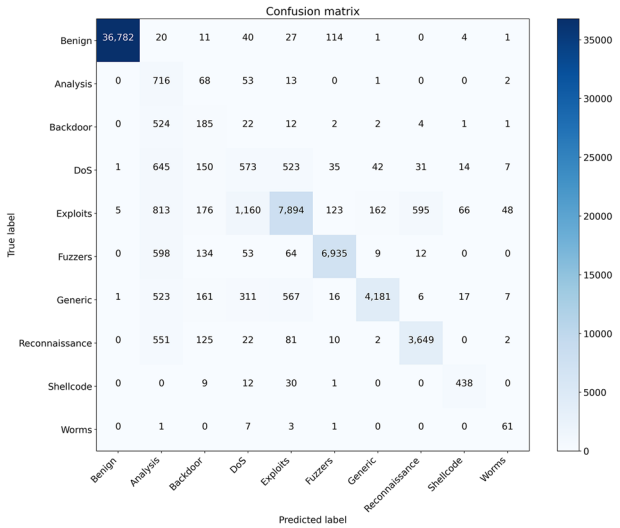
The average classification precision of the proposed method for different attack types on three datasets reached 98.247%, 98.805%, and 91.688%, respectively, the highest among all schemes. Benefiting from the strong ability to extract detailed features, the misclassification of the proposed method and BERT is significantly less than that of LSTM and BiLSTM. The average F1-score of the proposed method for different attack types classification on the CSE-CIC-IDS2018 dataset reached 98.230%, which was 1.218% higher than BERT, and it also exhibited specific improvements on the other two datasets, demonstrating that the proposed method has improved performance in both precision and recall. Compared with BERT, BiLSTM, and LSTM, the versions with CGAN boost precision, recall, and F1-score. Nevertheless, the proposed method combines the strong feature extraction ability of BERT and the strong generalization ability of GAN. The advantages of these two aspects further reduce misclassification and false negatives and improve F1-score.



(a) Confusion matrix for CSE-CIC-IDS2018.



(b) Confusion matrix for NF-ToN-IoT-v2.



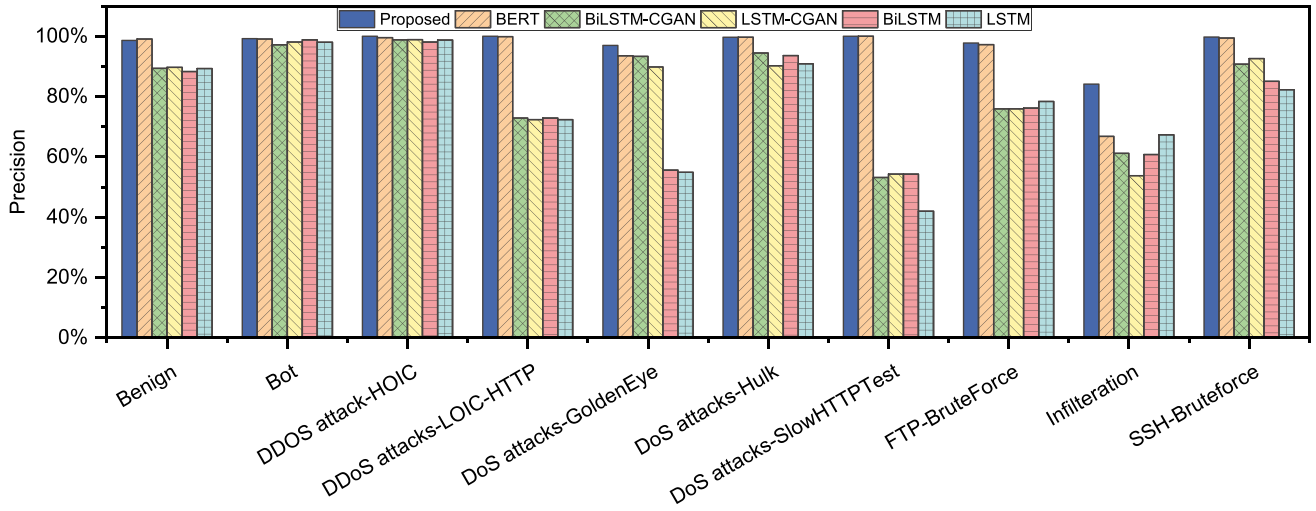
(c) Confusion matrix for NF-UNSW-NB15-v2.

Fig. 5 Confusion matrix of the proposed method on three datasets

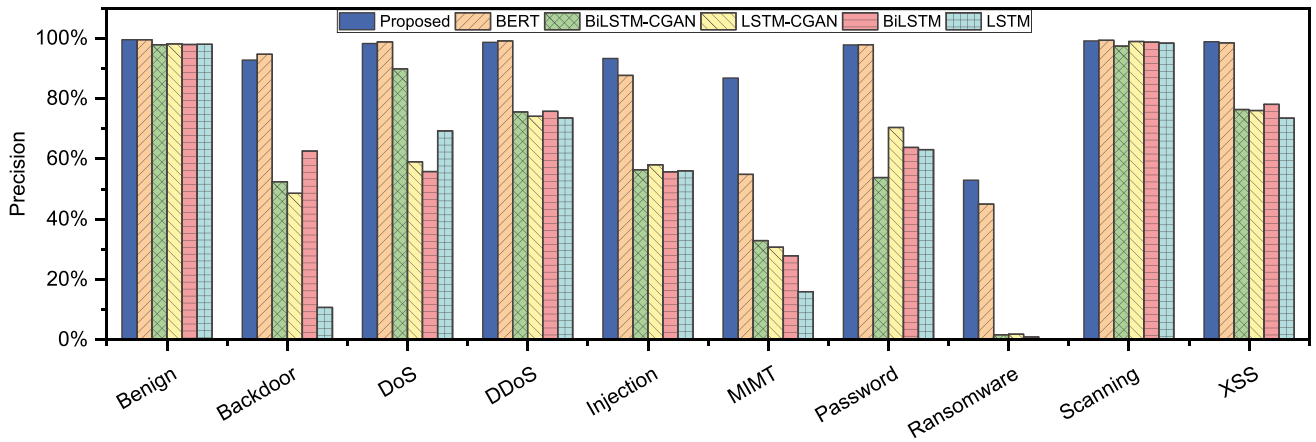
Figure 5 shows the confusion matrix of the proposed method for multi-class detection on these three datasets. Most of the samples in the CSE-CIC-IDS2018 and NF-ToN-IoT-v2 datasets were concentrated on the diagonal of the matrix, indicating that almost all types of network traffic were correctly identified. For the more category-imbalanced NF-UNSW-NB15-v2 dataset, the proposed method could also successfully detect most of the attack types. Notably, within the CSE-CIC-IDS2018 dataset, the proposed method achieved a precision of over 95% for the DoS attack types - SlowHTTPTest and DoS attacks-GoldenEye, which account for only about 2% each. There are only 90, 36, and 18 samples of Backdoor, MITM, and Ransomware attacks in the NF-ToN-IoT-v2 dataset, and only 73 and 490 samples of Worms and Shellcode attacks in the NF-UNSW-NB15-v2 dataset. The proposed method achieved a recall rate of 80% or higher for these types of attacks. Our method alleviates the imbalance problem and detects various attacks with high accuracy.

5.2 Precision, recall and F1-score for different attack types

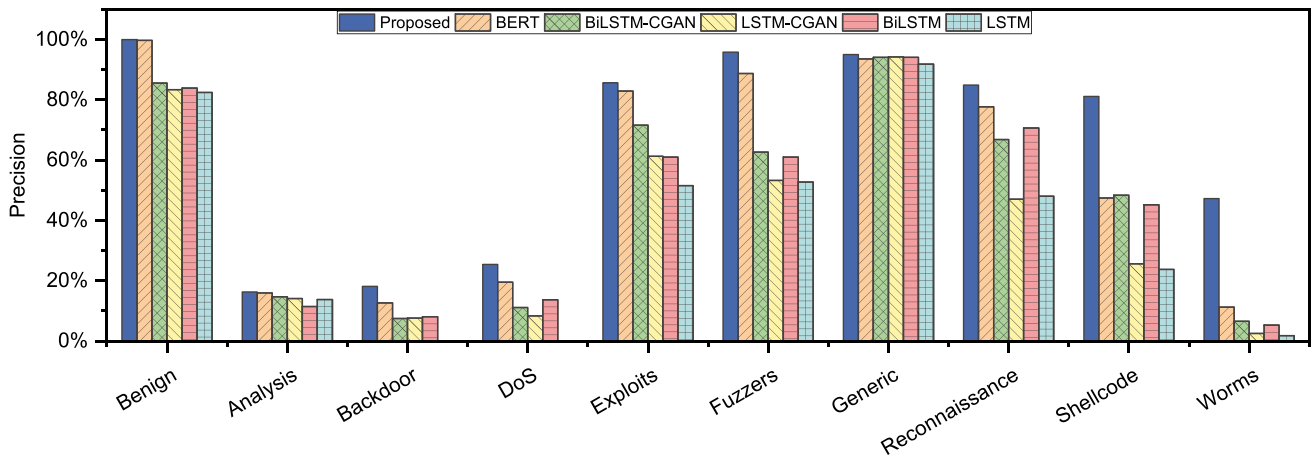
Figures 6, 7 and 8 show the precision, recall, and F1-score of different methods, respectively. The proposed method has almost the highest precision compared to the other five solutions. Each piece of network traffic corresponds to multiple statistical features, and these statistical features cover



(a) Precision of all classes for CSE-CIC-IDS2018.

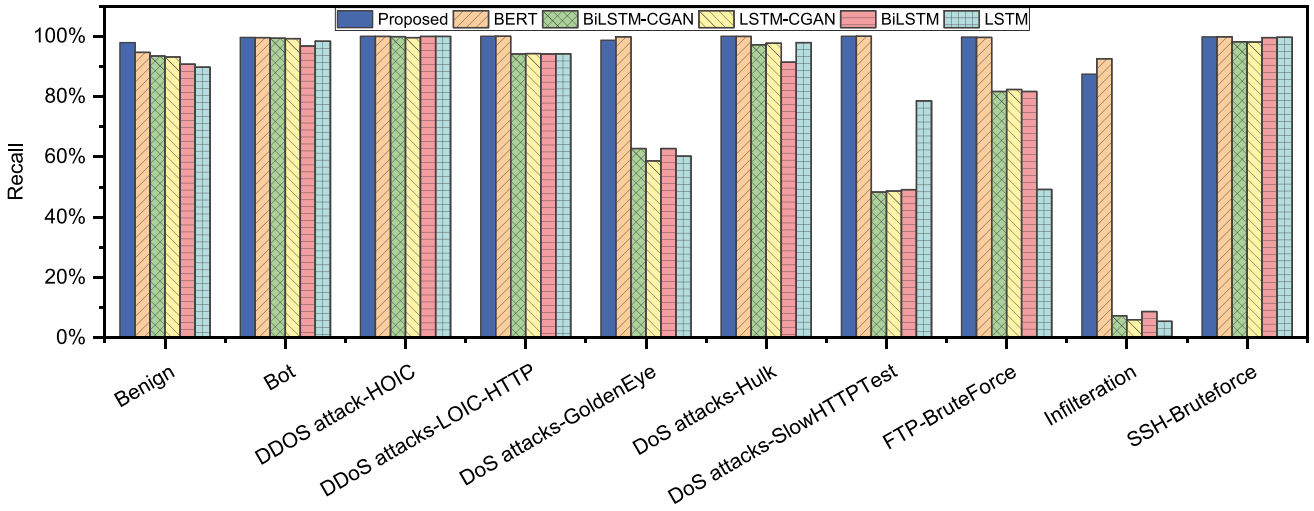


(b) Precision of all classes for NF-ToN-IoT-v2.

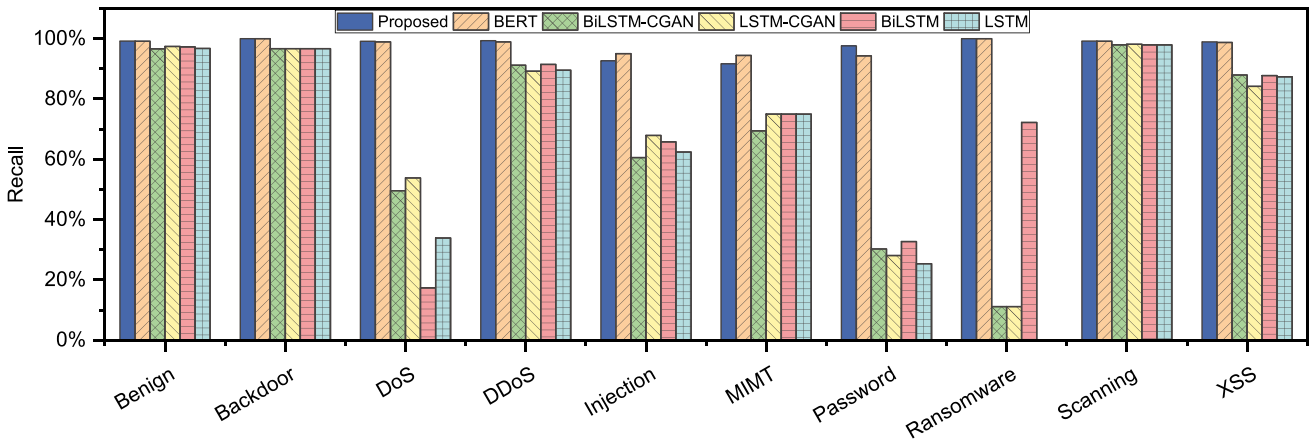


(c) Precision of all classes for NF-UNSW-NB15-v2.

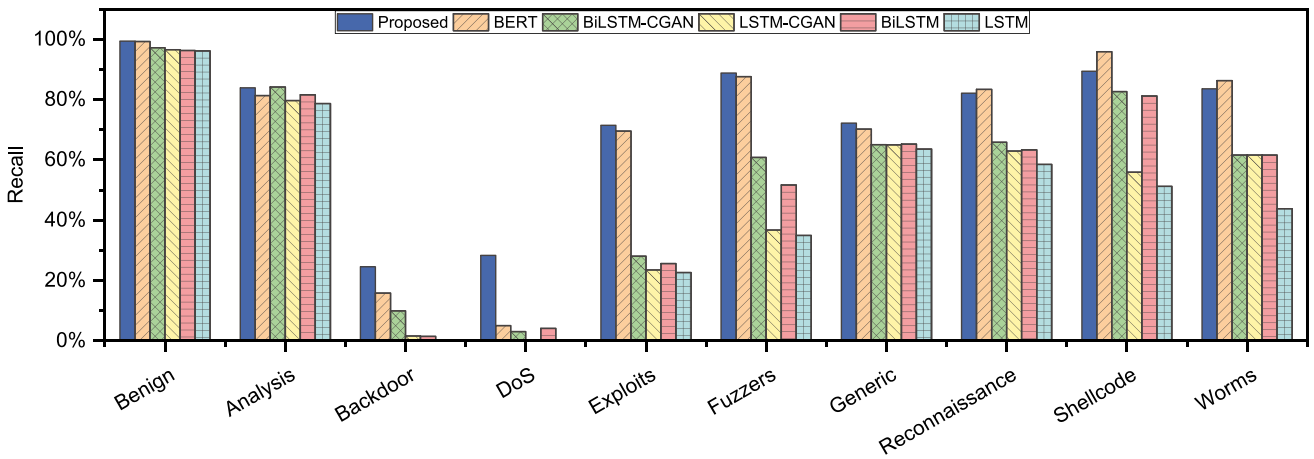
Fig. 6 Precision for benign and individual attack classes on three datasets



(a) Recall of all classes for CSE-CIC-IDS2018.

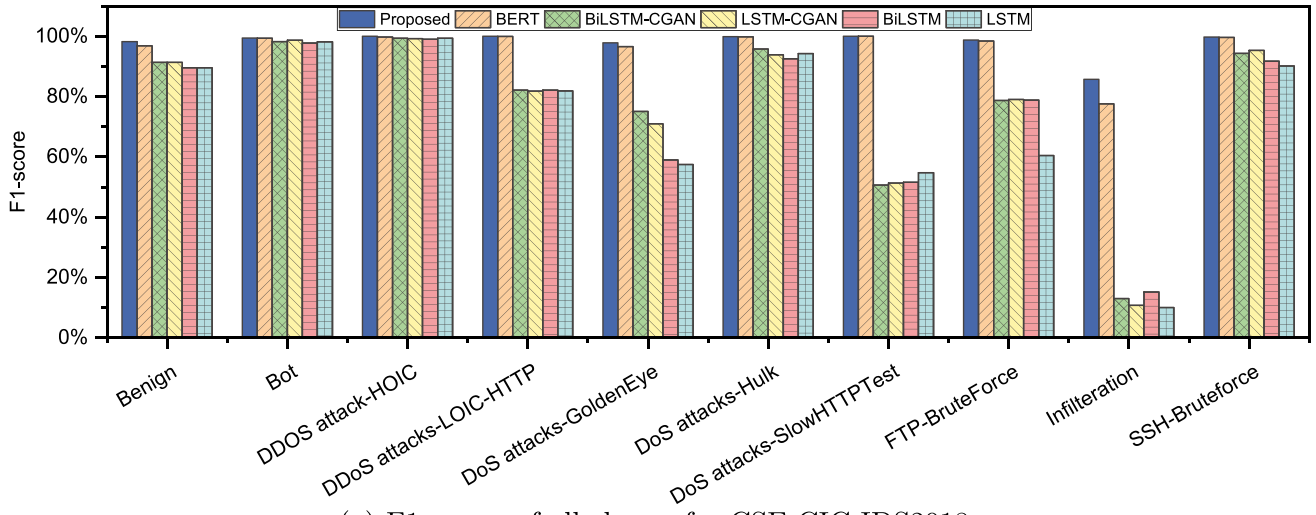


(b) Recall of all classes for NF-ToN-IoT-v2.

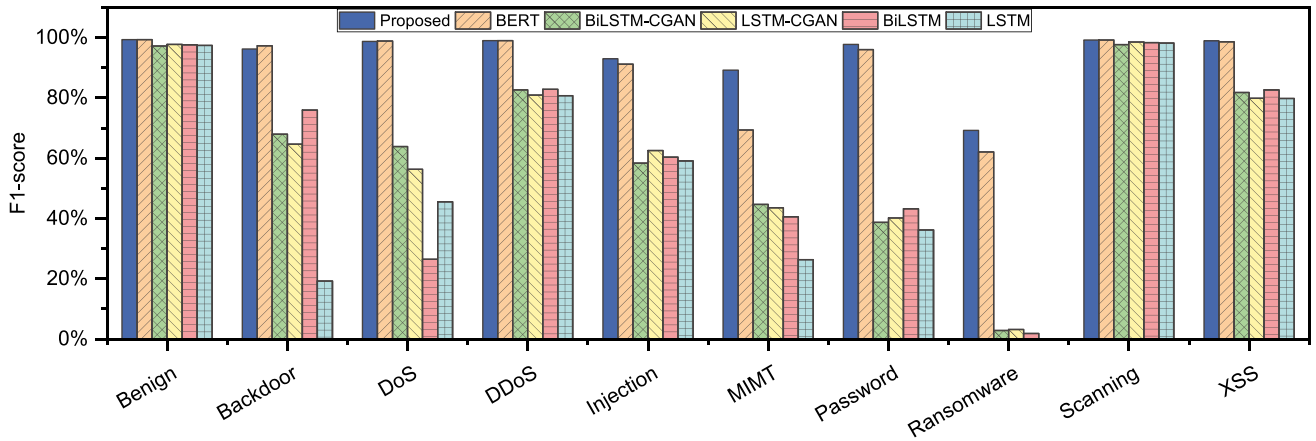


(c) Recall of all classes for NF-UNSW-NB15-v2.

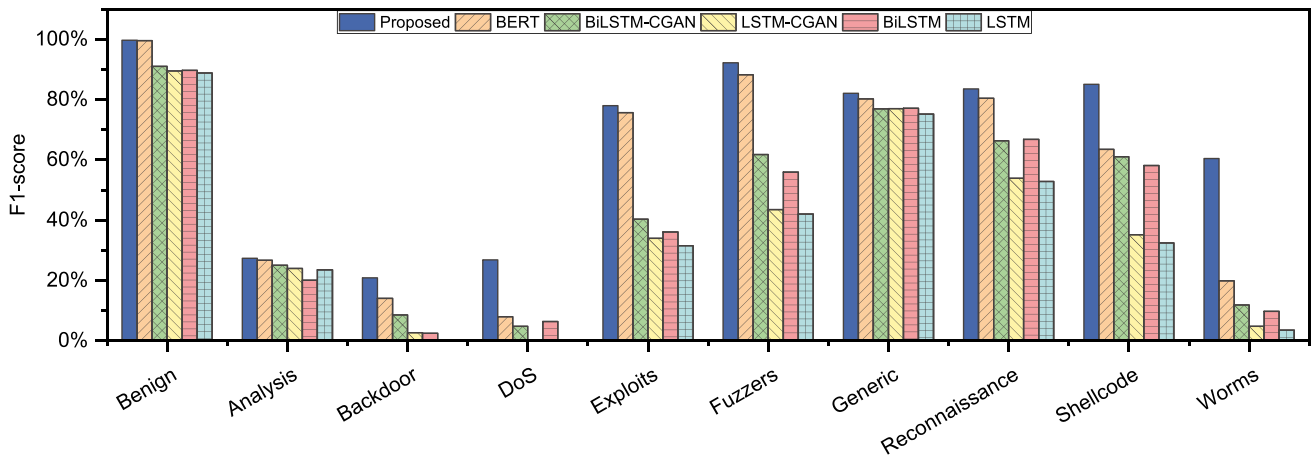
Fig. 7 Recall for benign and individual attack classes on three datasets



(a) F1-scores of all classes for CSE-CIC-IDS2018.



(b) F1-scores of all classes for NF-ToN-IoT-v2.



(c) F1-scores of all classes for NF-UNSW-NB15-v2.

Fig. 8 F1-scores for benign and individual attack classes on three datasets

network traffic characteristics information. Each statistical feature corresponds to a word with a specific meaning. The intrinsic relationship among statistical features is equivalent to the contextual relationship in a sentence.

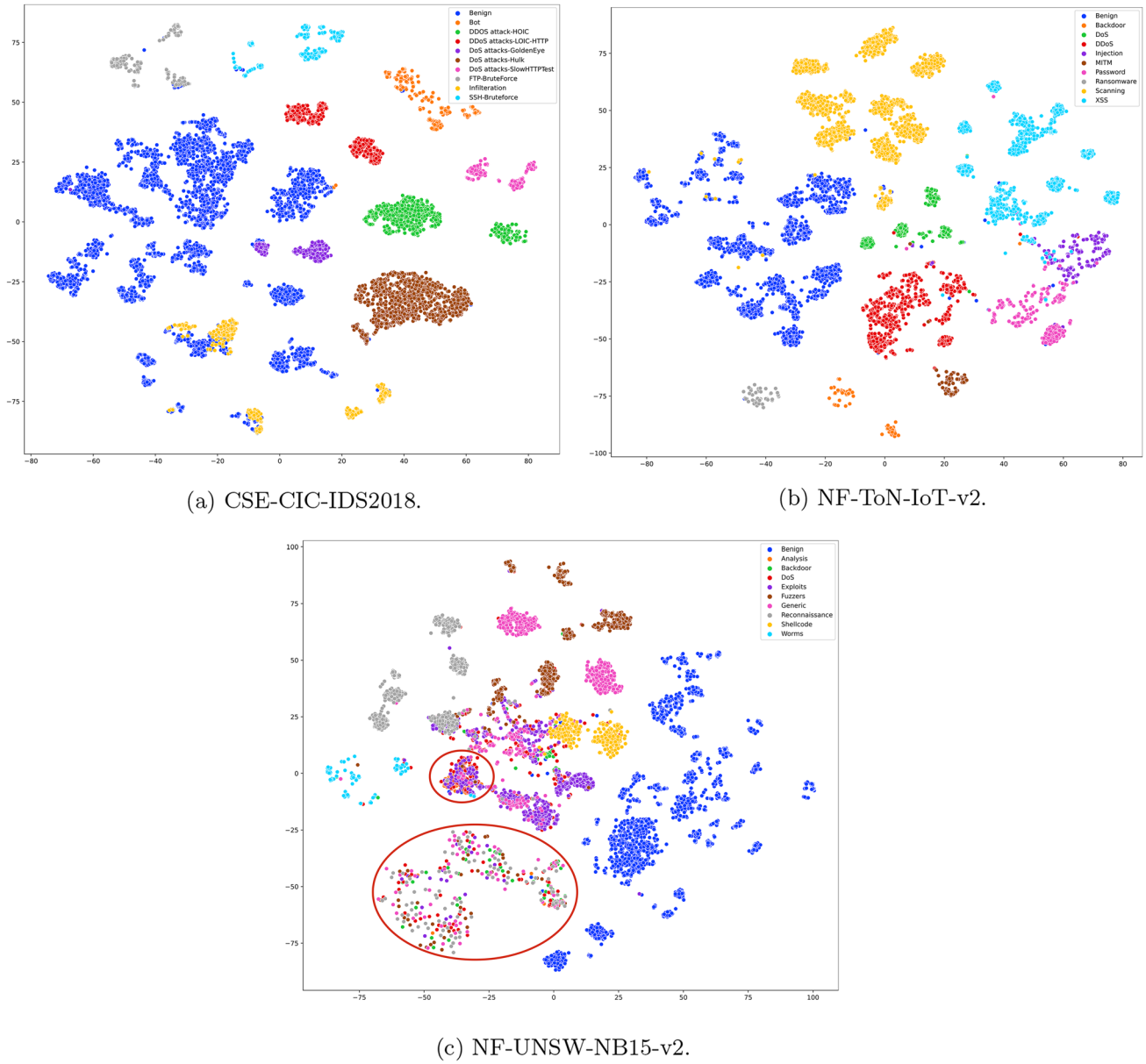
As shown in Fig. 6, for the CSE-CIC-IDS2018 dataset, among all ten categories, BiLSTM and LSTM had five classes with classification precision more significant than 80%, including Benign, Bot, DDOS attack-HOIC, DoS attacks-Hulk, and SSH-Bruteforce. The first type is the benign data with the most significant proportion of the test set, and the last four types are the attack types with the most data entries in the test set. Since DoS attacks-GoldenEye and DoS attacks-SlowHTTPTest are the two types with the fewest data among all traffic types, they are more challenging to identify. The LSTM had less than 55% precision for these two types. LSTMs can learn to compensate for the minimum time lag of long discrete time steps by enforcing a constant time step [16]. In this way, LSTM pays attention to the forward dependencies of network traffic features. Due to the forward nature of time series, from the perspective of contextual relations, LSTM mainly emphasizes the forward dependencies of network traffic features, ignoring the backward dependencies. BiLSTM, as a bidirectional LSTM, pays attention to the forward and backward dependencies of network features, making up for the shortcomings of LSTM to a certain extent. BiLSTM is composed of bidirectional LSTMs, covering both forward and backward dependencies. Due to the enhanced feature-capturing ability, BiLSTM has higher precision in identifying network attacks than traditional LSTM. Even with the two types of attacks (i.e., DoS attacks-GoldenEye and DoS attacks-SlowHTTPTest) with fewer data, the detection precisions of BiLSTM hover between 50% and 60%, still unimpressive. Unlike BERT, based on the self-attention mechanism, LSTM and BiLSTM suffer from the inherent problem of information decay when dealing with long sequences. Thus, they are prone to misclassification for some types of network attacks with petite proportions and complex hidden feature information.

Compared with BiLSTM, the precision of the proposed method and BERT in identifying Infilteration attacks in the CSE-CIC-IDS2018 dataset increased by 23.391% and 6.036%, respectively, reaching 84.119% and 66.764%. The precision of the proposed method for the remaining categories was all above 95%. The Ransomware and MITM attacks in the NF-ToN-IoT-v2 dataset have only 18 and 36 samples, respectively, the two attacks with a minor proportion. Compared with BiLSTM, the proposed method improved the precision of these two attack types by 52.004% and 59.007%, respectively, and the BERT increased by 44.063% and 27.004%, respectively. In the more imbalanced NF-UNSW-NB15-v2 dataset, the proposed method improved the precision for the most minor represented attack types, Worms and Shellcode, by 41.993% and 35.884%, respectively. In contrast, BERT increased them by 5.936% and 2.2%, respectively. The BERT model can significantly alleviate the problem that some attack types were previously difficult to identify, especially the classes with a small proportion.

From Fig. 7, the proposed method and BERT had a recall of 94% and above in 9 categories except for Infilteration in the CSE-CIC-IDS2018 dataset. The recall for ten traffic types in the NF-ToN-IoT-v2 dataset was above 90%. For the NF-UNSW-NB15-v2 dataset with more unbalanced categories, the proposed method and BERT demonstrated significant improvements in recall across all traffic types compared to BiLSTM and LSTM. Notably, when detecting the least represented Worms and Shellcode attack types, both models achieved recall improvements ranging from 6.735% to 44.694%, with both exceeding 80%. A notable phenomenon is that the recall rate of the BERT model is higher than the precision rate for most network attack classes, but the recall rate for the largest Benign class is lower than the precision rate. Due to the emphasis on checking the attack class, in some cases, BERT may classify the Benign class as an attack class. Hence, the recall in the Benign class with the most significant proportion is lower than that of some attack classes with a small proportion rate. Nevertheless, BERT achieved over 94% recall for normal traffic on all three datasets, and the proposed method further improved Benign recall on CSE-CIC-IDS2018 by 3.259%.

Figure 8 shows the F1-scores of different methods in detecting benign and malicious traffic. The proposed method achieves higher F1-scores than baselines for most attack categories, especially compared to BiLSTM and LSTM. On CSE-CIC-IDS2018, notable improvements are observed in detecting DDoS, GoldenEye, SlowHTTPTest, FTP-BruteForce, and Infiltration attacks, with 17.85%−75.73% increases. Similarly, on NF-ToN-IoT-v2, the proposed method excels in detecting Backdoor, DoS, Injection, MITM, Password, and Ransomware attacks, improving by 20.27%−76.99%. On NF-UNSW-NB15-v2, significant gains are attained in detecting Backdoor, DoS, Exploits, Fuzzers, Shellcode, and Worms, with 18.40%−56.91% higher F1-scores. Many of these attacks have limited samples or high concealment.

Although the advantages over standalone BERT are reduced, the proposed model still achieves noticeable improvements for certain attack types. For highly concealed attacks like Infiltration in CSE-CIC-IDS2018 and MITM in NF-ToN-IoT-v2, as well as Worms, Shellcode, Backdoor, Analysis and DoS in NF-UNSW-NB15-v2, BERT struggles with false positives. In contrast, the proposed model increases F1-scores by 8.18% for Infiltration, 19.80% for MITM, and up to 40.52% for the aforementioned attacks in NF-UNSW-NB15-v2. The



(a) CSE-CIC-IDS2018.

(b) NF-ToN-IoT-v2.

(c) NF-UNSW-NB15-v2.

Fig. 9 Visualization of high-dimensional spatial representations extracted by BERT on three datasets: Each graph contains 10000 points

integration of BERT and CGAN enables further performance gains, especially for previously challenging attack types.

The proposed method has improved the detection performance. The room for performance improvement becomes very limited for categories where F1-score reaches 99%, and the value approaches the upper bound. Even with improved performance, the contribution to the remaining identification classes becomes weak. From Fig. 8, the detection performance for those types with F1-score over 99% is already hard to improve. Except for those categories, the proposed method achieves a higher F1-score than BERT in the rest of the classes, indicating that the proposed method outperformed BERT in both precision and recall.

5.3 Visualization

To gain a more intuitive understanding of the performance of the proposed method, we randomly extracted 10000 samples from the dataset and visualized the classification results. BERT maps network traffic to different locations in the high-dimensional space according to traffic feature information, clarifying the boundaries among different types of network traffic. Figure 9 shows the distribution of various types of network traffic after the high-dimensional space representation extracted by BERT is reduced to two-dimensional space. From Fig. 9, the boundaries of different network traffic categories are relatively straightforward, which means the proposed framework can achieve accurate

classification. As shown in Fig. 6, the proposed method and BERT significantly improved the precision for small-scale attack types compared to other methods. For instance, in the CSE-CIC-IDS2018 dataset, the proposed method and BERT outperformed other methods in detecting Infiltration and DoS attacks-GoldenEye type.

Similarly, in the NF-ToN-IoT-v2 dataset, they demonstrated better performance in identifying Ransomware, MITM, and Backdoor attacks. Additionally, in the NF-UNSW-NB15-v2 dataset, the proposed method and BERT showed notable improvements in detecting Worms and Shellcode attacks. The boundaries of these types are more explicit in Fig. 9.

In the classification visualization diagram of the NF-UNSW-NB15-v2 dataset, categories with unclear boundaries were marked by red circles. Analysis, Backdoor, and DoS attack types are mixed without clear boundaries. Considering the confusion matrix on this dataset, some Analysis attacks were misclassified as Backdoor and DoS attacks, and the same misclassification phenomenon existed for Backdoor and DoS attacks. Due to the high similarity and strong concealment of some attack categories in the NF-UNSW-NB15-v2 dataset, baseline methods such as BiLSTM and LSTM are challenging to identify. Although the proposed method of BERT-enhanced feature extraction improves the detection performance, it still has a high false positive rate.

6 Conclusion

In this paper, we have proposed a pre-trained language model enhanced CGAN for multi-class network intrusion detection. The framework leverages CGAN to augment minority attack data for balancing the training set and improving generalization. By embedding BERT in the CGAN discriminator, more informative features can be extracted to identify network attacks. Through adversarial training, the BERT-enhanced discriminator enables the generator to produce higher-quality samples close to the actual data distribution, thereby boosting intrusion detection performance. Extensive experiments on three benchmark datasets demonstrate that the proposed method achieves superior overall results compared to baseline approaches.

Further analysis reveals that distinguishing attacks with similar characteristics or high concealment remains challenging, such as Analysis, Backdoor, and DoS in the NF-UNSW-NB15-v2 dataset. As large language models possess robust semantic understanding and generation capabilities, applying them to analyze network traffic data could be a promising direction, which we leave for future work.

Acknowledgements The authors gratefully acknowledge the financial assistance provided by the National Natural Science Foundation of China, the Natural Science Foundation of Jiangsu Province, and other research projects.

Author contributions Fang Li, Hang Shen, and Jieai Mai wrote the main manuscript text. Tianjing Wang, Yuanfei Dai, and Xiaodong Miao provided guiding ideas and suggestions. All authors reviewed the manuscript.

Funding This work was supported in part by National Key R &D Program of China under Grant 2021YFB2012301, the National Natural Science Foundation of China under Grants 61502230, 61501224, and 62202221, the Natural Science Foundation of Jiangsu Province under Grant BK20201357 and BK20220331, the Six Talent Peaks Project in Jiangsu Province under Grant RJFW-020.

Data availability Data sharing is not applicable to this article as no new data were created or analyzed in this study.

Declarations

Ethical approval and consent to participate This article contains no studies with human participants or animals performed by any of the authors.

Consent for publication All authors agree to publish the paper and related research results.

Conflict of interest We declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper. We declare that there is no financial interest/personal relationship which may be considered as potential competing interests.

References

1. Chou D, Jiang M (2021) A Survey on Data-driven Network Intrusion Detection. ACM Comput Surv (CSUR) 54(9):1–36
2. Kilincer IF, Ertam F, Sengur A (2021) Machine learning methods for cyber security intrusion detection: Datasets and comparative study. Comput Netw 188:107840
3. Gamage S, Samarabandu J (2020) Deep learning methods in network intrusion detection: A survey and an objective comparison. J Netw Comput Appl 169:102767
4. Mummadi A, Yadav BMK, Sadhwika R, Shitharth S (2021) An appraisal of cyber-attacks and countermeasures using machine learning algorithms. In International Conference on Artificial Intelligence and Data Science, pages 27–40
5. Wang H, Gu J, Wang S (2017) An effective intrusion detection framework based on SVM with feature augmentation. Knowl-Based Syst 136:130–139
6. Koc L, Mazzuchi TA, Sarkani S (2012) A network intrusion detection system based on a Hidden Naïve Bayes multiclass classifier. Expert Syst Appl 39(18):13492–13500
7. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) SMOTE: Synthetic Minority Over-sampling Technique. J Artif Intell Res 16:321–357
8. Jia H, Liu J, Zhang M, He X, Sun W (2021) Network intrusion detection based on IE-DBN model. Comput Commun 178:131–140
9. Wu T, Fan H, Zhu H, You C, Zhou H (2022) Huang X (2022) Intrusion detection system combined enhanced random forest with smote algorithm. EURASIP J Adv Signal Process 1:1–20

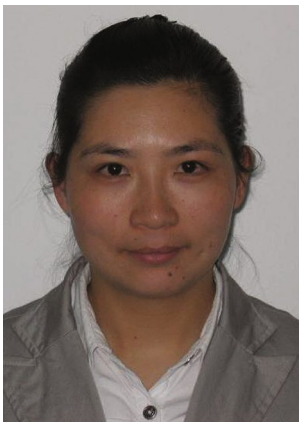
10. Mikhail JW, Fossaceca JM, Iammartino R (2019) A semi-boosted nested model with sensitivity-based weighted binarization for multi-domain network intrusion detection. ACM Trans Intell Syst Technol (TIST) 10(3):1–27
11. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. Adv Neural Inf Proces Syst 27
12. Lee J, Park K (2021) GAN-based imbalanced data intrusion detection system. Pers Ubiquit Comput 25(1):121–128
13. Lin Z, Shi Y, Xue Z (2022) IDSGAN: Generative adversarial networks for attack generation against intrusion detection. In Pacific-Asia Conference on Knowledge Discovery and Data Mining, pages 79–91
14. Ding H, Chen L, Dong L, Fu Z, Cui X (2022) Imbalanced data classification: A KNN and generative adversarial networks-based hybrid approach for intrusion detection. Futur Gener Comput Syst 131:240–254
15. He X, Chen Q, Tang L, Wang W, Liu T (2022) Cgan-based collaborative intrusion detection for uav networks: A blockchain-empowered distributed federated learning approach. IEEE Internet Things J 10(1):120–132
16. Hochreiter S, Schmidhuber J (1997) Long Short-Term Memory. Neural Comput 9(8):1735–1780
17. Lin SZ, Shi Y, Xue Z (2018) Character-level intrusion detection based on convolutional neural networks. In International Joint Conference on Neural Networks (IJCNN), pages 1–8
18. Aydın H, Orman Z, Aydın MA (2022) A long short-term memory (LSTM)-based distributed denial of service (DDoS) detection and defense system design in public cloud network environment. Comput Secur 118:102725
19. Huang Z, Xu W, Yu K (2015) Bidirectional LSTM-CRF Models for Sequence Tagging. arXiv preprint arXiv:1508.01991
20. Roy B, Cheung H (2018) A deep learning approach for intrusion detection in internet of things using bi-directional long short-term memory recurrent neural network. In International Telecommunication Networks and Applications Conference (ITNAC), pages 1–6
21. Kim J, Kim J, Thu HLT, Kim H (2016) Long short term memory recurrent neural network classifier for intrusion detection. In International Conference on Platform Technology and Service (PlatCon), pages 1–5
22. Althubiti SA, Jones EM, Roy K (2018) LSTM for Anomaly-Based Network Intrusion Detection. In International Telecommunication Networks and Applications Conference (ITNAC), pages 1–3
23. Imrana Y, Xiang Y, Ali L, Abdul-Rauf Z (2021) A bidirectional LSTM deep learning approach for intrusion detection. Expert Syst Appl 185:115524
24. Shitharth S, Satheesh N, Kumar BP, Sangeetha K (2021) IDS detection based on optimization based on WI-CS and GNN algorithm in SCADA network. Architectural Wireless Networks Solutions and Security Issues 247–265
25. Ling C, Zhao X, Lu J, Deng C, Zheng C, Wang J, Chowdhury T, Li Y, Cui H, Zhao T et al (2023) Beyond one-model-fits-all: A survey of domain specialization for large language models. arXiv preprint arXiv:2305.18703
26. Devlin J, Chang MW, Lee K, Toutanova K (2018) BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv preprint arXiv:1810.04805
27. Yin J, Tang MJ, Cao Jinli, Wang Hua (2020) Apply transfer learning to cybersecurity: Predicting exploitability of vulnerabilities by description. Knowl-Based Syst 210:106529
28. Lee Y, Kim J, Kang P (2021) LAnoBERT: System log anomaly detection based on bert masked language model. arXiv preprint arXiv:2111.09564
29. Alkhatib N, Mushtaq M, Ghauch H, Danger JL (2022) CAN-BERT do it? controller area network intrusion detection system based on bert language model. In IEEE/ACS 19th International Conference on Computer Systems and Applications (AICCSA), pages 1–8
30. Mirza M, Osindero S (2014) Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784
31. Douzas G, Bacao F (2018) Effective data generation for imbalanced learning using conditional generative adversarial networks. Expert Syst Appl 91:464–471
32. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. Adv Neural Inf Process Syst 30
33. Kingma DP, Ba J (2014) Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980
34. Salem M, Taheri S, Yuan JS (2018) Anomaly Generation Using Generative Adversarial Networks in Host-Based Intrusion Detection. In IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), pages 683–687

Fang Li received the B.S. degree in Computer Science from Nanjing Tech University, Nanjing, China, where she has continued pursuing her M.S. degree since September 2022. Her research interests include large language models and cybersecurity.

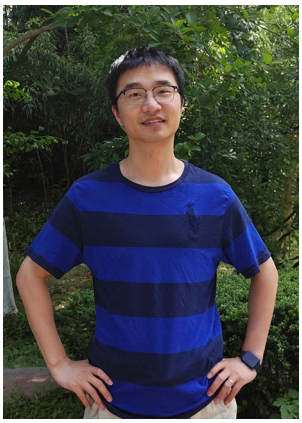
Hang Shen received the Ph.D. degree (with honors) in Computer Science from the Nanjing University of Science and Technology. He worked as a Full-Time Postdoctoral Fellow with the Broadband Communications Research (BBCR) Lab, ECE Department, University of Waterloo, Waterloo, ON, Canada, from 2018 to 2019. He is currently an Associate Professor at the Department of Computer Science and Technology, Nanjing Tech University, Nanjing, China. His research interests involve space-air-ground integrated networks, network slicing, blockchain, and cybersecurity. He has published research papers in prestigious international journals and conferences, including the IEEE Transactions on Mobile Computing, IEEE Transactions on Broadcasting, IEEE Transactions on Vehicular Technology, *Elsevier Journal of Systems Architecture*, and IEEE

ICC. He is an Associate Editor for the IEEE Access and an Academic Editor of the *Mathematical Problems in Engineering*. He was a TPC member of the Annual International Conference on Privacy, Security and Trust (PST) 2021. He is a senior member of CCF, a member of IEEE, and an executive committee member of ACM Nanjing Chapter.

Jieai Mai received the B.S. degree (with honors) in Computer Science from Nanjing Tech University, Nanjing, China, in 2022 and is currently an M.S. student at the Institute of Artificial Intelligence, Xiamen University, Xiamen, China. Her research interests are in physics-informed neural networks and network intrusion detection.



Tianjing Wang holds a B.Sc. in Mathematics at the Nanjing Normal University in 2000, an M.Sc. in Mathematics at Nanjing University in 2002, and a Ph.D. in Signal and Information Systems at the Nanjing University of Posts and Telecommunications in 2009. From 2011 to 2013, she was a Full-Time Postdoctoral Fellow with the School of Electronic Science and Engineering, Nanjing University of Posts and Telecommunications. From 2013 to 2014, she was a Visiting Scholar with the ECE Department at the State University of New York at Stony Brook. She is currently an Associate Professor with the Department of Communication Engineering at Nanjing Tech University. Her research interests include distributed machine learning for cellular V2X communication networks, blockchain, and network security. She has published research papers in prestigious international journals and conferences, including the IEEE Transactions on Mobile Computing, IEEE Transactions on Broadcasting, and IEEE ICC. She is a member of IEEE and a member of CCF.



Yuanfei Dai received the Ph.D. degree from the College of Computer and Data Science, Fuzhou University in 2021. He is currently a lecturer with the College of Computer and Information Engineering, Nanjing Tech University. His research interests include knowledge acquisition, knowledge graph representation, and bioinformatics.

Xiaodong Miao received the B.S. and Ph.D. degrees in mechanical engineering from Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2008 and 2013, respectively. He is currently an Associate Professor of mechanical engineering at Nanjing Tech University. His research interests include sensors and signal processing, machinery condition monitoring, fault diagnostics, and artificial intelligence applications in network intrusion detection. He is a member of IEEE.