Expert Systems With Applications 238 (2024) 122198

Contents lists available at ScienceDirect

Expert Systems With Applications

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

HDA-IDS: A Hybrid DoS Attacks Intrusion Detection System for IoT by using semi-supervised CL-GAN

Sifan Li [a], Yue Cao [a,*], Shuohan Liu [b], Yuping Lai [c], Yongdong Zhu [d], Naveed Ahmad [e]

[a] School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China
[b] Warwick Manufacturing Group, University of Warwick, Coventry CV4 7AL, UK
[c] School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing 100876, China
[d] Institution of Intelligent System, Zhejiang Lab, Hangzhou 311000, China
[e] Department of Computer Science, Prince Sultan University, Riyadh 11586, Saudi Arabia

ARTICLE INFO

Keywords:
Internet of Things
Intrusion Detection System
Machine learning
Generative adversarial network

ABSTRACT

In recent years, the application of the internet of things (IoT) in areas such as intelligent transportation, smart cities, and the industrial internet has become increasingly widespread. As a crucial supporting infrastructure, IoT devices are utilized in various fields to construct IoT networks. However, due to the inherent limitations of IoT devices, such as limited computing resources and low memory capacity, security concerns have become increasingly prominent. Among these concerns are Denial-of-Service (DoS) and botnet attacks, which are difficult to prevent due to their large-scale and covert nature. To address these challenges, this paper proposes a Hybrid DoS Attack Intrusion Detection System (HDA-IDS) that combines signature-based detection with anomaly-based detection to effectively identify both known and unknown DoS/botnet attacks. Additionally, this paper introduces a novel anomaly-based detection model called CL-GAN. It integrates CNN-LSTM with GAN to establish a baseline for normal behavior and detect malicious traffic. In contrast to other semi-supervised models, the CL-GAN exhibits superior accuracy, as well as shorter training and testing times, in detecting DoS and botnet attacks. In addition, experimental results demonstrate that the HDA-IDS outperforms other IDSs in detecting DoS and botnet attacks. When tested on datasets such as NSL-KDD, CICIDS2018, and Bot-IoT, the HDA-IDS achieved an average of 5% overall improvement superior performance in terms of accuracy, precision, recall, and F1-Score compared to other works. These results highlight the effectiveness of the proposed system in addressing security issues in IoT networks, and presents a general framework that addresses the challenge of large-scale attacks constructed through the dissemination of false information.

1. Introduction

The rapid advancement of the Internet of Things (IoT) has propelled the widespread adoption of diverse applications within smart cities. In the present era, the integration of devices such as smart home gadgets, Internet of Vehicles (IoV) components, and industrial machinery with the internet has become an inexorable trend. However, owing to resource limitations in hardware and a deficient security infrastructure, these devices are susceptible to cyber intrusions (Sarjan, Ameli, & Ghafouri, 2022). On one hand, limited by low cost, low power consumption and limited computing resources of IoT devices, it is difficult to configure security mechanisms (encryption algorithms) for IoT devices (Vishwakarma & Jain, 2020). On the other hand, the openness of IoT makes devices, networks and platforms easy to be attacked or controlled. Devices are directly exposed to local area network, Internet and other networks. Attackers can utilize simple tools to analyze confidential information and privacy information stored in the same type of devices, thus launching large-scale attacks.

In this context, the susceptibility of individual devices to vulnerabilities can serve as a pivotal point of compromise, posing a significant threat to IoT security. Exploiting the security frailties of such devices, malevolent actors can manipulate software to establish control, thus rendering devices as zombie machines. Moreover, the consequences encompass data loss, system malfunctions, and even the compromise of sensitive personal data (Chowdhury, Sen, Goswami, Purkait, & Saha, 2023). Given the inadequate security fortification across a multitude of IoT devices, they remain exposed to network-layer attacks (Khanday, Fatima, & Rakesh, 2023). Particularly deleterious are Denial of Service (DoS) and botnet attacks, distinguished by their extensive reach

* Corresponding author.
E-mail addresses: sifan.li@whu.edu.cn (S. Li), yue.cao@whu.edu.cn (Y. Cao), shuohan.liu@warwick.ac.uk (S. Liu), laiyp@bupt.edu.cn (Y. Lai), zhuyd@zhejianglab.com (Y. Zhu), nahmed@psu.edu.sa (N. Ahmad).

https://doi.org/10.1016/j.eswa.2023.122198
Received 26 May 2023; Received in revised form 2 October 2023; Accepted 15 October 2023
Available online 28 October 2023

and surreptitious nature. These attacks can compromise IoT devices, inducing abnormal behavior. Subsequent control by hackers could lead to catastrophic consequences. Establishing an Intrusion Detection System (IDS) within the IoT network represents a proactive defense strategy. Furthermore, a Machine Learning (ML)-based IDS proves effective in discerning network traffic patterns and identifying malicious activity. The IDS approach is classified into signature-based detection and anomaly-based detection, each contingent on distinct detection methodologies.

In recent years, the ascendancy of Machine Learning (ML) technology has propelled the evolution of IDSs. Many IDSs have been combined with relevant ML techniques to enhance both detection efficiency and precision (Lampe & Meng, 2023). Notably, supervised learning algorithms such as Support Vector Machines (SVM), Gradient Boosting Decision Tree (GBDT), and eXtreme Gradient Boosting (XGBoost) have found substantial application in the IDS domain. Although SVM, GBDT, and XGBoost represent foundational learning techniques, they exhibit inherent limitations. Here, SVM struggles with large-scale training datasets and multi-classification problems, XGBoost faces challenges with ultra-high dimensional feature data, and GBDT is prone to overfitting while disregarding feature correlations within datasets. Hence, applying the Stacking can solve the shortcomings of basic learning models. Stacking is a basic method of data science, and it can integrate GBDT, XGBoost and other algorithms. Meanwhile, the stacking framework integrates different algorithms, and it makes full use of the observation of data by different algorithms to learn from each other and optimize the results.

Moreover, deep learning technologies are also widely applied in IDS, such as Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM). However, CNN and LSTM have their own shortcomings. CNN is not suitable for learning time series, especially for time series sensitivity problems.[1] Here, LSTM is usually suitable, because it has a certain memory effect. Instead, LSTM networks have longer processing times. Therefore, there is a considerable body of prior research that employs CNN-LSTM as a central component within IDS for the primary purpose of identifying malicious traffic (Alferaidi et al., 2022). In these studies, the CNN-LSTM combines the strengths of both CNN and LSTM, effectively leveraging temporal characteristics of time series data to deeply excavate data features.

Nevertheless, the CNN-LSTM is ineffective when dealing with large-scale false samples or unbalanced data (the discrepancy in the quantity of data among different categories is significant). To facilitate data augmentation, prior research has proposed the utilization of Generative Adversarial Networks (GAN) for data manipulation (Ferdowsi & Saad, 2019). Inspired by GAN and CNN-LSTM, we combine GAN with CNN-LSTM to establish a normal behavior pattern and identify the false data. Moreover, our approach enables the detection of sophisticated attacks and the ability to adapt to emerging threats. By leveraging expert knowledge and machine learning, our research contributes to the advancement of intelligent systems for cyber security.

In view of these, there are a small number of IDSs focusing on using GAN to detect attacks at present (Chauhan & Shah Heydari, 2020; Seo, Song, & Kim, 2018). These IDSs in literature employ GAN for intrusion detection, one in the context of in-vehicle attack detection (Seo et al., 2018) and the other in generating adversarial distributed DoS attacks (Chauhan & Shah Heydari, 2020). Different from above works, our approach not only utilizes GAN networks to synthesize attack traffic for data augmentation, but also incorporates CNN-LSTM to enhance the detection capabilities of IDS in the IoT environment. Since anomaly-based detection lacks the detection of false data in traffic, the main contribution of this paper is as follows:

- A large number of ML based IDSs are proposed in literature to detect attacks. However, the detection efficacy of most IDSs for unknown attacks in the IoT domain is limited, especially concerning DoS and botnet attacks. This will threaten the security of IoT network when large-scale attacks occur on devices. Furthermore, in large-scale attacks, there is a significant presence of unknown attack threats that pose a security risk to the IoT. Aiming at this, we propose a Hybrid DoS Attacks Intrusion Detection System (HDA-IDS) by combining signature-based detection and anomaly-based detection. For known DoS and botnet attacks, the HDA-IDS applies Stacking technology to organize the results of multiple basic learners. For unknown/zero-day attacks, the HDA-IDS applies a new model to establish the behavior pattern.
- In contrast to existing literature on the application of GAN-based IDSs for the identification of unknown and zero-day attacks, our study introduces a novel GAN model, termed as CL-GAN. It integrates CNN-LSTM with GAN to establish a baseline for normal behavior and address the challenge of detecting unknown or zero-day attacks. The proposed model not only harnesses the intrinsic attributes of time-series and sequential data for enhanced data feature extraction, but also employs GAN methodology to discern fraudulent traffic messages. This approach solves the defects of CNN-LSTM in large-scale false data processing, and establishes a baseline of legitimate behavioral patterns, facilitating the differentiation of malicious traffic instances.
- Moreover, the CL-GAN employs a semi-supervised approach. The supervised model is trained using real datasets, while the unsupervised model is trained using fake data. Once the unsupervised model demonstrates stable performance, the fake data is amalgamated with real data and subsequently input to the supervised model. Compared to other semi-supervised models, the CL-GAN demonstrates superior accuracy and efficiency in detecting DoS and botnet attacks.

Next, Section 2 lists related works about IDS with comparative summarization. Section 3 shows the application scenario of HDA-IDS and the model of HDA-IDS. Section 4 presents the experiment setup, datasets, evaluation metrics and results. Section 5 finally concludes this paper.

2. Related work

2.1. Signature-based IDS

The signature-based IDS analyzes known attacks to extract their distinguishing features and patterns as signatures. These signatures are compared with network traffic to detect intrusion. For instance, Alaba, Maitanmi, and Ajayi (2019) utilized the Stacking Ensemble technique including Random Forest (RF), Naive Bayes (NB) classification models and SVM to detect network traffic. In another study, a MLP-based network IDS was developed by Rosay, Carlier, and Leroux (2020) to detect cyber-attacks within IoT environments. The performance of this model was assessed on the CICIDS2017 dataset, achieving a high accuracy rate exceeding 99%. The suitability of six classical ML algorithms for detecting botnet attacks on the CICIDS2017 dataset was evaluated by Aswal, Dobhal, and Pathak (2020). In their comparative study, recurrent Neural Network (RNN), LSTM and Gate Recurrent Unit (GRU) are combined in Hai and Nam (2021) to detect intrusion on CICIDS2017/CICIDS2018 datasets. In addition, Gamage and Samarabandu (2020) trained Feed-forward Neural Network (FNN), autoencoder, Deep Belief Network (DBN) and LSTM to classify intrusion, and tested them on NSL-KDD, CICIDS2017 and CICIDS2018 datasets. Finally, Koroniotis, Moustafa, Sitnikova, and Turnbull (2019) generated the Bot-IoT dataset and gave a detailed description of designing the testbed configuration. The Bot-IoT dataset was evaluated the performance of network forensic methods by RNN/LSTM. A two-tier DDoS attack detection method, leveraging

[1] Time series sensitivity problems refer to issues or challenges that arise when analyzing and interpreting time series data, such as outliers, missing values, or changes in the underlying patterns over time.

information entropy and deep learning within the software-defined network (SDN) framework, was proposed in (Liu et al., 2022). Initially, the information entropy detection mechanism identifies potentially problematic components and ports in a broad context. Subsequently, a more detailed packet-level detection process is carried out using a CNN model to differentiate between regular and suspicious traffic.

2.2. Anomaly-based IDS

Anomaly-based IDS, also known as behavior-based IDS, utilizes ML methods to build the behavior pattern. Those patterns can identify normal and abnormal network traffic patterns. Karami (2018) employed a modified Self-Organizing Map (SOM) with a neural projection architecture to detect anomalies and attacks. Abdelmoumin, Rawat, and Rahman (2021) applied Stacking technique to create multiple learners, which were subsequently used to train single-learner models for Anomaly-based Machine Learning-enabled Intrusion Detection Systems (AML-IDS). The objective of utilizing AML-IDS is to improve the detection performance of single-learner models. The anomaly-based IDS proposed by Hodo et al. (2016) employed Artificial Neural Network (ANN) techniques to identify DoS/Distributed Denial of Service (DDoS) attacks within IoT networks. The application of this approach was evaluated using the DoS dataset from KDD 99. Dash (2017) proposed GSPSP-ANN, combined Grouping-Shuffling with Particle Swarm Optimization (GSPSO) to detect intrusion on the NSL-KDD dataset. The RNN-IDS, based on RNN, was proposed in Yin, Zhu, Fei, and He (2017) for intrusion detection, and tested on NSL-KDD dataset. Its multi-classification accuracy reached 81.29%. The ODM-ADS is an anomaly-based IDS proposed by Moustafa, Choo, Radwan, and Camtepe (2019), which consists of an adversarial statistical learning mechanism. Based on novel statistical learning, the IoTBoT-IDS was proposed to detect botnet attacks (Ashraf et al., 2021). It includes a lightweight statistical feature extraction model and a BMM-Correntropy method for detecting botnet attacks. An anomaly IDS that can identify unknown attacks within IoT networks was developed through the utilization of fog-edge collaborative analytics (Rahman et al., 2020). It solves the limitation imposed by centralized IDS on devices with limited resources, and semi-distributed and distributed methods are proposed.

2.3. Hybrid IDS

Although the signature-based IDS performs well in detecting known attacks, it will be helpless when faced with unknown attacks or new attacks. Meanwhile, the major disadvantage of anomaly-based IDS is that it produces false positives at a high detection rate. Instead, hybrid IDS integrates signature-based and anomaly-based techniques to enable the detection of both known and unknown attacks. Yang, Moubayed, and Shami (2021) proposed a Multitiered Hybrid Intrusion Detection System (MTH-IDS), which utilized multiple ML algorithms to identify attacks on both in-vehicle and external networks. Meanwhile, MTH-IDS includes new CL-KMeans based on K-Means to detect zero-day attacks. MTH-IDS was tested on CICIDS2017 and CAN-intrusion datasets. A new framework termed Hybrid Intrusion Detection System (H-IDS) was introduced by Vadursi, Ceccarelli, Duarte, and Mahanti (2016), which combines signature-based and anomaly-based detection techniques to detect DDoS attacks. In H-IDS, the anomaly-based detection utilized multidimensional Gaussian Mixture Models (GMMs), while the signature-based detector utilized SNORT. In literature (Khraisat, Gondal, Vamplew, Kamruzzaman, & Alazab, 2019), a new ensemble Hybrid Intrusion Detection System (HIDS) was introduced for detecting IoT attacks, which utilizes a C5 classifier and One Class SVM classifier. The Bot-IoT dataset was employed to evaluate the performance of this approach. In literature (Aydın, Zaim, & Ceylan, 2009), a hybrid IDS was developed by integrating Network Traffic Anomaly Detection (NETAD) and Packet Header Anomaly Detection (PHAD). This IDS can detect both known and unknown attacks. A hybrid IDS for the smart grid scenario was proposed to detect cyber attacks in literature (Pan, Morris, & Adhikari, 2015). It learns temporal state-based patterns encompassing power system scenarios, ranging from disturbances and normal control operations to instances of cyber attacks. Another hybrid learning model was proposed to detect unknown DoS/DDoS attacks in IoT networks (Nguyen & Le, 2023). This model belongs to semi-supervised model and uses nearest-neighbor ensembles (iNNE). It was evaluated on BoT-IoT, CIC-IDS-2017, and CIC-IDS-2018 datasets.

2.4. Motivation

As shown in Table 1, although there are a number of papers that apply hybrid IDS, these hybrid IDSs perform not well when they suffer from DoS and botnet attacks. This is mainly attributed to the presence of a substantial amount of false traffic in DoS and botnet attacks, which adversely affects the effectiveness of hybrid IDS in detecting such large-scale deceptive traffic. In contrast to existing studies, this paper introduces a novel hybrid IDS approach tailored for DoS and botnet attacks in the realm of IoT. Termed HDA-IDS, the proposed method combines signature-based IDS with anomaly-based IDS to effectively detect these types of attacks. On one hand, the signature-based detection component employs the Stacking technique to accurately identify and classify known attacks. On the other hand, the anomaly-based detection component utilizes a state-of-the-art CL-GAN model to detect unknown or zero-day attacks. The CL-GAN leverages the temporal and data series characteristics to perform in-depth analysis of data features, thereby enhancing data quality. Additionally, the CL-GAN harnesses GAN technology to discern and expose false traffic messages, establishing a reliable framework to distinguish normal and malicious traffic patterns.

3. Proposed HDA-IDS framework

The overall framework of HDA-IDS is shown in Fig. 1. Here, in this section, the HDA-IDS is described from five aspects: application scenarios, data pre-processing, feature selection, signature-based detection, and anomaly-based detection.

3.1. Application scenarios

Along with the popularization of IoT technology, a large number of devices are connected to the network. The IoT transmits data through diversified network connections, achieving the function of remote control and remote maintenance. Although the IoT has brought great convenience to people, the probability of being attacked is increasing. As shown in Fig. 2, IoT devices and network are vulnerable to botnet attacks and DoS attacks. In order to defend against these attacks, the proposed HDA-IDS is set in control center or gateway. Firstly, it achieves detecting traffic from the external network. Since most hackers utilize remote attacks (illegal access, impersonation attack, DoS attack, etc.) to gain control of IoT devices, traffic from external networks should receive extensive attention. Secondly, since hackers compromise the IoT network by invading devices entities (brute force, illegal access, etc.), the HDA-IDS needs to detect traffic from the internal network. Finally, once the HDA-IDS detects malicious traffic, it will send a warning to the control center, which is convenient for managers to deal with in time.

3.2. Data pre-processing

During data pre-processing, the realistic data is usually incomplete (missing significant attribute values), inconsistent (containing variations in codes or names), and susceptible to noise (errors) that can easily influence its accuracy and reliability. As the sources of data are different and the database is too large, data analysis results will be

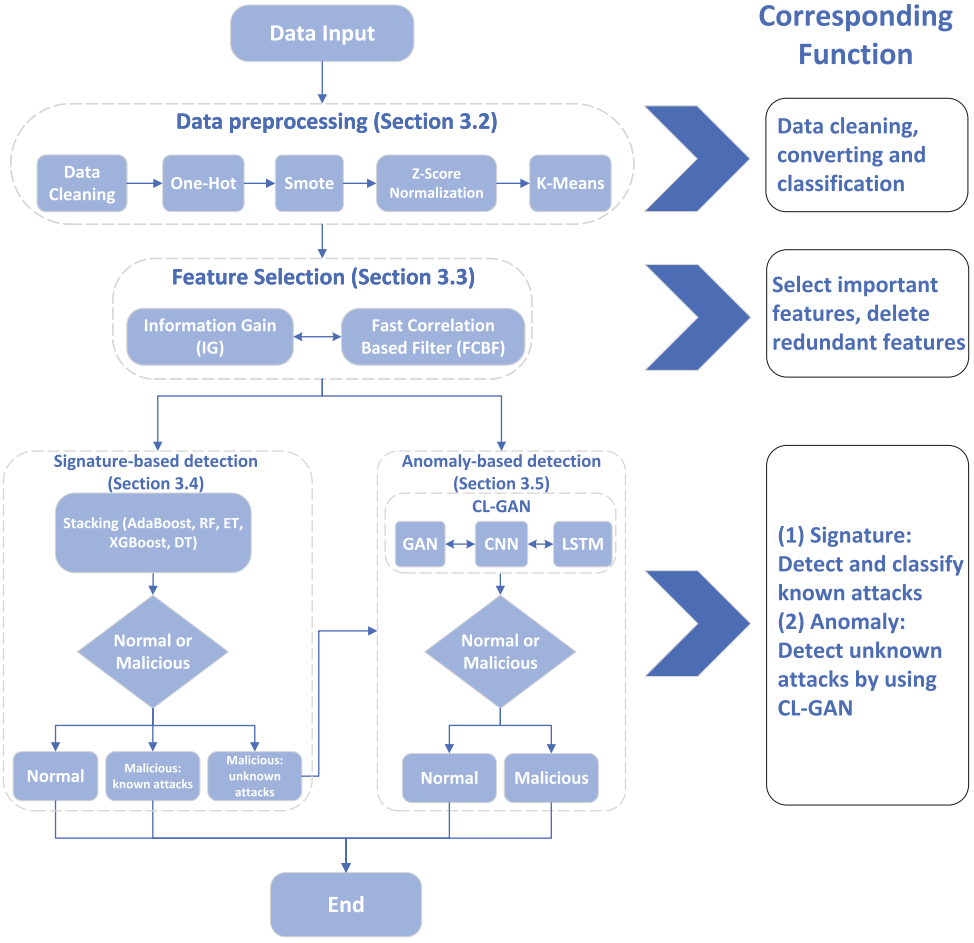

Fig. 1. The structure of HDA-IDS.

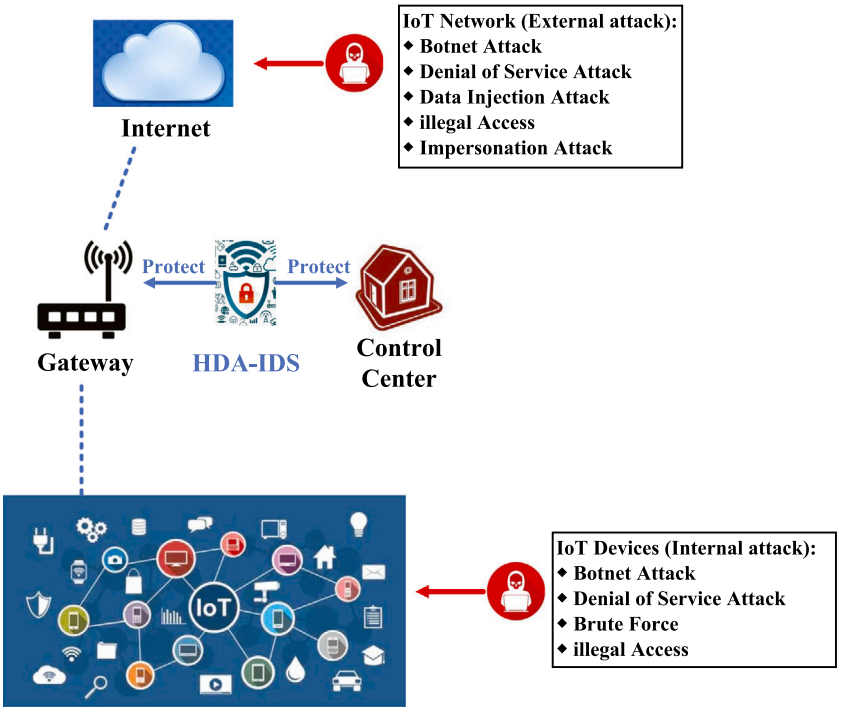

Fig. 2. HDA-IDS application scenarios.

Table 1
Comparison of related work.

Related work	Type of IDS	Datasets	Attack detection	Method
Alaba et al. (2019)	Signature-based IDS	NSL-KDD	DoS, R2L, Probe, U2R	Stacking
Rosay et al. (2020)	Signature-based IDS	CICIDS2017	Botnet, DDoS, DoS, FTP, Web attack, etc.	Multi-layer perceptron
Aswal et al. (2020)	Signature-based IDS	CICIDS2017	Botnet	NB, KNN, LR, LDA, CART, SVM
Hai and Nam (2021)	Signature-based IDS	CICIDS2017/2018	DoS, Botnet, FTP, Web attack, etc.	RNN, LSTM, GRU
Gamage and Samarabandu (2020)	Signature-based IDS	NSL-KDD, CICIDS2017/2018	DoS, Botnet, FTP, Web attack, etc.	FNN, DNN, Autoencoder, LSTM
Koroniotis et al. (2019)	Signature-based IDS	Bot-IoT	Botnet	RNN, LSTM
Liu et al. (2022)	Signature-based IDS	CICIDS2017	DDoS	CNN,Entropy
Karami (2018)	Anomaly-based IDS	NSL-KDD, UNSW-NB15, AAGM, VPN-nonVPN	Botnet, DoS, FTP, Webattack, etc.	Modified SOM
Abdelmoumin et al. (2021)	Anomaly-based IDS	ToN-IoT, UNSW-2018-IoT-Botnet	Botnet, DDoS, DoS, XSS, etc.	Stacking
Hodo et al. (2016)	Anomaly-based IDS	KDD-99	DDoS, DoS	ANN
Dash (2017)	Anomaly-based IDS	NSL-KDD	DoS, U2R, R2L	GSPSO-ANN
Yin et al. (2017)	Anomaly-based IDS	NSL-KDD	DoS, U2R, R2L	RNN
Moustafa et al. (2019)	Anomaly-based IDS	NSL-KDD, UNSW-NB15	Backdoors, DoS, U2R, R2L	ODM-ADS
Ashraf et al. (2021)	Anomaly-based IDS	Kitsune, ISCX, IoT network intrusion dataset	Botnet	IoTBoT-IDS
Rahman et al. (2020)	Anomaly-based IDS	AWID	Impersonation attacks	SAE, SVM, CFS, J48, OneR, and MLP
Yang et al. (2021)	Hybrid IDS	CICIDS2017, CAN-intrusion datasets	Cyber attacks on in-vehicle and external networks	Stacking, CL-Kmeans
Vadursi et al. (2016)	Hybrid IDS	DARPA 2000 dataset, A commercial bank's dataset	DDoS	GMMs, SNORT
Khraisat et al. (2019)	Hybrid IDS	Bot-IoT dataset	IoT attacks	C5, SVM
Aydın et al. (2009)	Hybrid IDS	DARPA, IDEVAL	Probe, DOS, R2L, etc.	Hybrid IDS
Pan et al. (2015)	Hybrid IDS	Simulated real, power system scenarios	Cyber-attacks in power system	Hybrid IDS
Nguyen and Le (2023)	Hybrid IDS	CICIDS2017/2018,Bot-IoT	DoS/DDoS	SOCNN,LOF,iNNE
Proposed	Hybrid IDS	CICIDS2018, Bot-IoT, NSL-KDD	DoS, Botnet	Stacking, CL-GAN(GAN, CNN, LSTM)

affected by low-quality data. Therefore, before the formal training of datasets, the steps such as data cleaning, converting and classification should be operated. Here, data pre-processing realizes the process of correcting damaged, inaccurate or inapplicable records from data. In the HDA-IDS, data pre-processing includes data cleaning, One-Hot, smote, Z-score normalization and K-Means.

3.2.1. Data cleaning

Data cleaning is applied to detect and correct (or delete) inaccurate or missing value data in the dataset. In the HDA-IDS, data cleaning includes deleting repetitive data and incomplete data (i.e. many features of these data have been lost). Additionally, the different values of "label" feature are replaced with numbers.

3.2.2. One-Hot

Classification features are often dealt with in data processing, but a lot of feature values are discrete and disordered. Hence it is usually necessary to digitize its features. The One-Hot technique (Rodríguez, Bautista, Gonzalez, & Escalera, 2018) initially involves the mapping of a classification value to an integer value. Subsequently, each integer value is translated into a binary vector representation. For each non-digital feature, if it has M values, then it becomes M binary features after One-Hot coding. In the NSL-KDD (Tavallaee, Bagheri, Lu, & Ghorbani, 2009) and CICIDS2018 (Sharafaldin, Lashkari, & Ghorbani, 2018) dataset, the feature named protocol_type in the dataset need to be processed with One-Hot coding. For example, there are three types of protocols: TCP, UDP and ICMP. Therefore, this column of features can be replaced with three equivalent columns of features (protocol_type_tcp, protocol_type_udp and protocol_type_icmp). The specific process is shown in Fig. 3:

3.2.3. Smote

There is often a serious imbalance between different categories of datasets. If there is a serious imbalance in datasets, the predicted conclusions are often biased towards classes which have a large number of samples. For example, Linear Regression (LR) is not suitable for dealing with category imbalance, because it tends to judge samples into most categories. Although it can achieve high accuracy, it has a low recall rate. To solve this problem, we utilize the Smote technology (Chawla, Bowyer, Hall, & Kegelmeyer, 2002), as an improved oversampling algorithm based on random sampling, to balance the datasets. The fundamental premise of the Smote algorithm involves the analysis and emulation of minor sample categories (attack samples) followed by the addition of synthesized samples to the dataset. This integration generates a more balanced representation between the classes within the original dataset.

First, a sample y is selected from category with minor samples. Secondly, according to the sampling ratio o, y_o are neighbors of y and they are randomly selected. Finally, the new samples are randomly synthesized between y_o and y in turn, a new synthetic instance y_{new} is denoted by

$$y_{new} = y + \beta \times (y_o - y) \tag{1}$$

Where β represents a random number in the range of (0, 1).

3.2.4. Z-Score normalization

In data pre-processing, many methods require samples to meet certain standards. In the HDA-IDS, it usually needs to process the data with large size gap or even different magnitude. Processing these data will affect the result and spend much time. Hence the HDA-IDS applies

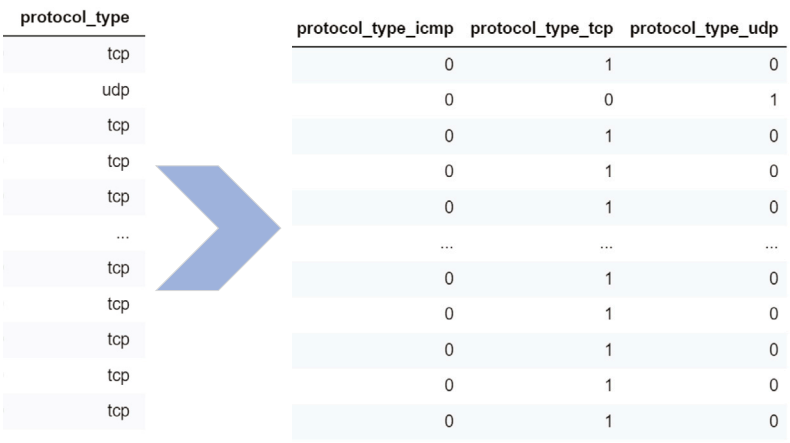

Fig. 3. One-Hot code.

the Z-Score normalization (Fei, Gao, Lu, & Xiang, 2021) to process these data with excessive size gap. The Z-Score normalization is utilized to standardize data scales and dimensions that may differ across different datasets. This method involves scaling data to uniform data intervals and ranges to limit the impact of scale discrepancies on the HDA-IDS. Besides, Z-Score has the advantage of easy calculation. Each of normalization value h is denoted by

$$h = \frac{a - \mu}{\sigma} \tag{2}$$

where a represents the original value, μ is the mean of the feature values, and σ is the standard deviation of feature values.

3.2.5. K-Means

To enhance the training speed and reduce the complexity of model training, it is necessary to sample the data. In this system, a clustering sampling algorithm based on K-Means (MacQueen, 1965) is adopted to process the dataset and obtain a highly representative subset. The core task of K-Means is to find out k optimal centroids, and minimize the sum of squares for distances between all data points and the corresponding centroids of clusters.

$$min \sum_{i=1}^{k} \sum_{B \in CL_i} \left(CL_i - B\right)^2 \tag{3}$$

where CL_i is the cluster center of i, and B is the cluster data point.

3.3. Feature selection

Datasets contain many redundant or useless features, even if removing these features will not lead to the loss of information. For example, there are 72 features in the CICIDS2018 dataset and 43 features in the NSL-KDD dataset, but not all features have an impact on traffic classification. Therefore, it is important to select those features highly related to the results and delete irrelevant or redundant features. The feature selection improves the accuracy of HDA-IDS and reduces the running time. The HDA-IDS utilizes Information Gain (IG) and Fast Correlation-Based Filter Solution (FCBF) algorithms to calculate and sort the importance of features. Then, the HDA-IDS selects the higher-importance features, which have a great influence on the classification results.

3.3.1. Information Gain (IG)

IG is calculated based on information entropy. It indicates the level in which information alleviates uncertainty, and selects features by sorting the IG. During feature selection, the target variable is applied as information U and the feature variable is applied as information V. By sorting the IG, the order of features is determined, so as to select features. The IG between the information U and V is denoted by

$$IG\left[U, V\right] = Ent\left[U\right] - Ent\left[U|V\right] \tag{4}$$

where $Ent\left[U\right]$ indicates the uncertainty before sending out the information U, and $Ent\left[U|V\right]$ indicates the average uncertainty of information U that still exists upon receiving V, that is, posterior uncertainty.

3.3.2. Fast Correlation-Based Filter Solution (FCBF)

FCBF is a fast filtering feature selection algorithm based on Symmetrical Uncertainty (SU) (Yu & Liu, 2003). FCBF can identify redundancy among features without paired correlation analysis. FCBF includes the following specific steps:

Firstly, it needs to calculate the correlation $SU\left[U, V\right]$ between each feature U and the target V. The calculation formula is as follows:

$$SU\left[U, V\right] = \frac{2 \cdot IG\left[U, V\right]}{Ent\left[U\right] + Ent\left[V\right]} \tag{5}$$

where $Ent\left[V\right]$ indicates the uncertainty before sending out the target V.

Secondly, the features with correlation degree greater than the preset threshold δ are selected. In order to obtain an appropriate correlation threshold, δ is optimized by Bayesian Optimization - Gaussian Process (BO-GP). This utilizes verification accuracy as the objective function of Hyper-parameter Optimization (HPO). Thirdly, the $SU\left[U, V\right]$ is arranged in descending order. The correlation $SU\left[U, W\right]$ between each feature U and all other features W smaller than $SU\left[U, V\right]$ are calculated in turn. Fourthly, the feature W with $SU\left[U, W\right]$ greater than $SU\left[W, V\right]$ is deleted, and finally the feature subset $S\left(s_1, s_2, s_3 ... s_n\right)$ is obtained.

3.4. Signature-based detection

Signature-based intrusion detection is commonly approached as a matching and classification problem, utilizing existing signatures for identification. In the context of HDA-IDS, the differentiation between normal and malicious network traffic data is achieved through multi-classification techniques. Specifically, supervised ML technology is employed to effectively leverage existing signatures and accurately classify samples.

The HDA-IDS applies ensemble learning (Stacking) (Džeroski & Ženko, 2004) to classify datasets. The Stacking method comprises two stages of models. In the first stage, the base model (also known as the level-0 model) is responsible for processing the original training set as input. The second stage is composed of the meta-model (also known as the level-1 model).

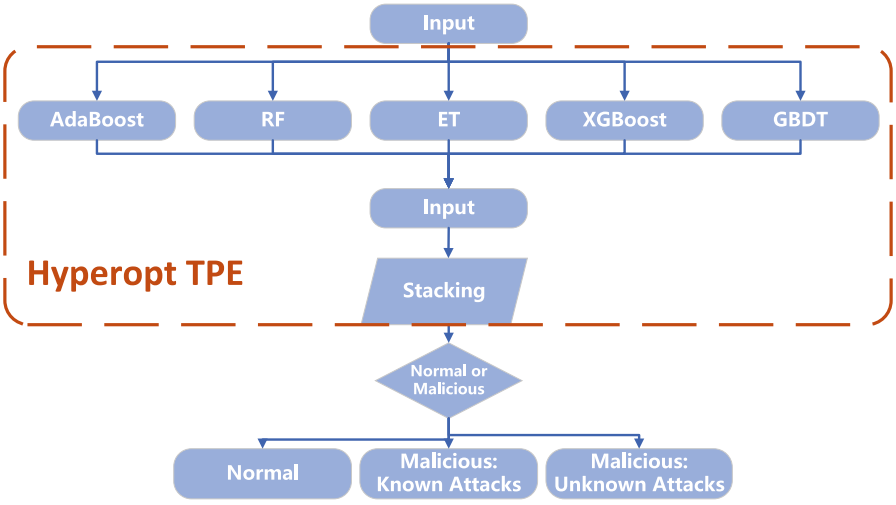

Fig. 4. Signature-based detection.

As shown in Fig. 4, signature-based detection applies Adaptive Boosting (AdaBoost), RF, Extra Trees (ET), XGBoost and GBDT to train the same dataset as the first-layer base model. Then it constructs the output of previous training base model into a training set. This training set is applied as input to train the Stacking model. The HDA-IDS applies Stacking to integrate the benefits of each model. Moreover, as Stacking is relatively stable (Shunmugapriya & Kanmani, 2013), the HDA-IDS provides better generalization performance than a single model.

The signature-based detection applies the Hyperopt-based Tree-structured Parzen Estimator (TPE) to optimize the hyperparameters of the five tree-based ML algorithms (Yang & Shami, 2020). As one of the most popular Bayesian optimizers, Hyperopt integrates many optimization algorithms including random search, simulated annealing and TPE (Lima, Ferreira Junior, & Oliveira, 2021). Furthermore, Hyperopt is a commonly applied optimizer to implement the TPE method. Actually, compared with Bayesian optimization based on Gaussian process, TPE based on Gaussian mixture model achieves better results with higher efficiency in most cases (Garrido-Merchán & Hernández-Lobato, 2020).

After Stacking, the signature-based detection will label traffic prediction results. In Fig. 4, the normal traffic and malicious traffic matching the known attacks will be output, while the malicious traffic of unknown attacks will be handed over to anomaly-based detection for processing.

3.5. Anomaly-based detection

3.5.1. The model of CNN-LSTM

The CNN-LSTM combines the advantages of CNN and LSTM models to improve the accuracy of malicious traffic prediction. The advantage of CNN is that it can extract effective features from data (Li, Zhang, & Wang, 2021). Meanwhile, the advantage of LSTM is that it discovers the interdependence of data in time series and detects the best pattern suitable for related data (Hewamalage, Bergmeir, & Bandara, 2021).

The structure of CNN-LSTM is shown in Discriminator (Fig. 7). According the output of FCBF S, it will be sent to convolutional layer:

$$L_n = LeakyRelu\left(w_n * s_n + b\right) \tag{6}$$

Where $LeakyRelu$ is the Leaky ReLU function, L_n is the output features (a total of n features) of CNN, w_n represents weight. b is deviation parameters. The L_n will be sent to the LSTM layer later.

In the CNN-LSTM, the LSTM is a modification of RNN, and its core concepts are cell state and "gate" structure. Cell state can record the path of information transmission to pass information in sequence. It can be regarded as the "memory" of network. As shown in Fig. 5, it mainly includes three parts: forget gate (f_t), input gate (i_t), and output gate (o_t). C_t is t_{th} cell state, and h_t is t_{th} hidden state of LSTM. tanh is the activation function that normalizes the content to (−1, 1).

3.5.2. The model of GAN

GAN is a deep learning model, which mainly includes two parts: the generator and discriminator of corresponding neural network. As shown in Fig. 6, the generator produces data according to the input random vector, and the discriminator judges whether this data is real or machine-generated, so as to judge whether the data is "false data" made by the generator. On the one hand, GAN can learn the distribution of data well, so as to extract the features of untrained data well. On the other hand, it can produce high-quality samples, which are more stable and easy to train. Therefore, GAN can realize data enhancement to improve the performance of classifier. Since GAN has Back Propagation (BP) characteristic, the classification model based on GAN has higher recognition accuracy and stability than the common neural network classification model.

The training process of a GAN usually involves the alternate training of two networks: the discriminator network and the generator network. They are trained alternately until a Nash equilibrium is reached. Both discriminator model D and generator model G adopt multilayer perceptron. GAN defines a noise $p_z(\boldsymbol{x})$ as a prior, which is applied to learn the probability distribution p_g of the generator model G on the training data x. $G(\boldsymbol{z})$ means mapping the input noise z into data. The probability that x comes from the real data distribution p_{data} instead of p_g is defined as $D(\boldsymbol{x})$. As a result, the objective function of optimization is expressed in the following min–max form:

$$\begin{aligned} GAN_{\text{loss}} &= \min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] \\ &\quad + \mathbb{E}_{\boldsymbol{z}\sim p_z(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))] \end{aligned} \tag{7}$$

$$\mathrm{G}_{\text{loss}} = -\log D(G(z)) \tag{8}$$

$$\mathrm{D}_{\text{loss}} = -\log D\left(x_{\mathrm{r}}\right) - \log(1 - D(G(z))) \tag{9}$$

Where GAN_{loss} is the adversarial loss of the whole GAN network. G_{loss} and D_{loss} represent the loss of the generator model G and the discriminant model D respectively.

- When updating the parameters of the discriminant model D: for the sample x from the real distribution p_{data}, it hopes that the output of $D(x)$ is close to 1, that is, $\log D(\boldsymbol{x})$ is higher. As for the data $G(\boldsymbol{z})$ generated by noise z, GAN needs $D(G(\boldsymbol{z}))$ to be as close as possible to 0 (D can distinguish between real and false data), so $\log(1 - D(G(\boldsymbol{z})))$ needs to be higher.

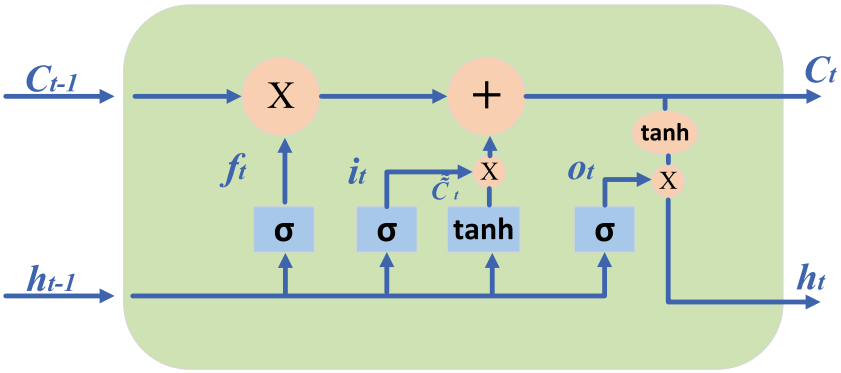

Fig. 5. The structure of LSTM cell.

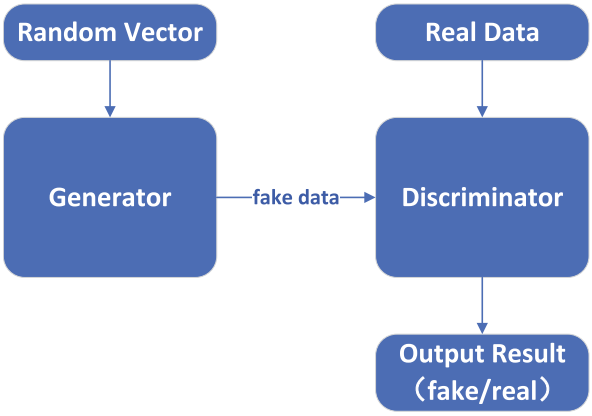

Fig. 6. The structure of GAN.

- When updating the parameters of the generator model G, it needs $G(z)$ to be the same as the real data as possible, that is, $p_g = p_{\text{data}}$. Therefore, it is hoped that $D(G(z))$ is as close to 1 as possible, that is, $\log(1 - D(G(z)))$ needs to be lower.

3.5.3. The model of CL-GAN

Algorithm 1 Define-Generator

Input: random vector(rv)
Output: *gen_model*
Reshape the *model.input*(rv)
Add *Dense* to the *model*
Add *Activation* to the *model*
Add *Conv2DTranspose* to the *model*
Add *BatchNormalization* to the *model*
Add *Conv2DTranspose* to the *model*
Add *Activation* to the *model*
Add *Conv2DTranspose* to the *model*
gen_out_layer ← *Activation*(*tanh*)
gen_model ← *model*(*gen_out_layer*)
fake_data ← *gen_model.fit*(rv)
Return *gen_model*, *fake_data*

Instead of paying attention to the threats of existing vulnerability database, the CL-GAN evaluates the traffic, analyzes the data flow, and checks the transmission requests from two aspects: transmission characteristics and protocols. Transmission characteristics includes flow duration, total packets in the forward direction, maximum size of packet in forward direction, number of forward packets per second, etc. Protocols refer to HTTPS, HTTP, SMTP, POP3, IMAP, SSH, and FTP. If necessary, the CL-GAN will restrict the network traffic, which will help prevent DoS and botnet attacks from expanding their harm.

As shown in Fig. 7, the CL-GAN mainly has three algorithms: In Algorithm 1, the generator convolutes the input random vector to generate a false data, which is reshaped and transmitted to the discriminator for subsequent neural network training.

Algorithm 2 Define-Discriminator

Input: *real_data*(rd), *fake_data*(fd)
Output: *d_model*, *c_model*
n ← Adjust according to rd, fd
for i in range n **do**
  Add *Conv1D* to the *model*
  Add *Maxpool* to the *model*
  Add *BatchNormalization*(*BN*) to the *model*
  Add *LeakyRelu* to the *model*
  Add *Dropout* to the *model*
end for
Add *LSTM* to the *model*
Add *Dense* to the *model*
Add *Dropout* to the *model*
Add *Flatten* to the *model*
Add *Dropout* to the *model*
Add *Dense* to the *model*
c_out_layer ← *Activation*(*sigmoid*)
c_model ← *model*(rd, *c_out_layer*)
d_out_layer ← *Lambda*()
d_model ← *model*(fd, *d_out_layer*)
Return *c_model*, *d_model*

After finishing fake data, it will be feed into discriminator with real data. The discriminator includes two models: (1) Supervised model (*c_model*) process real data to detect malicious traffic. (2) Instead, unsupervised model (*d_model*) is applied to discriminate the fake data generated by the generator. Once the loss and accuracy of unsupervised model detection tends to be stable, the generated fake data will be labeled, mixed with real data, and fed into supervised model. It helps to solve the imbalance of real data while it can strengthen the ability of supervised model to detect malicious traffic. As shown in Algorithm 2, the discriminator applies CNN to extract features from the input

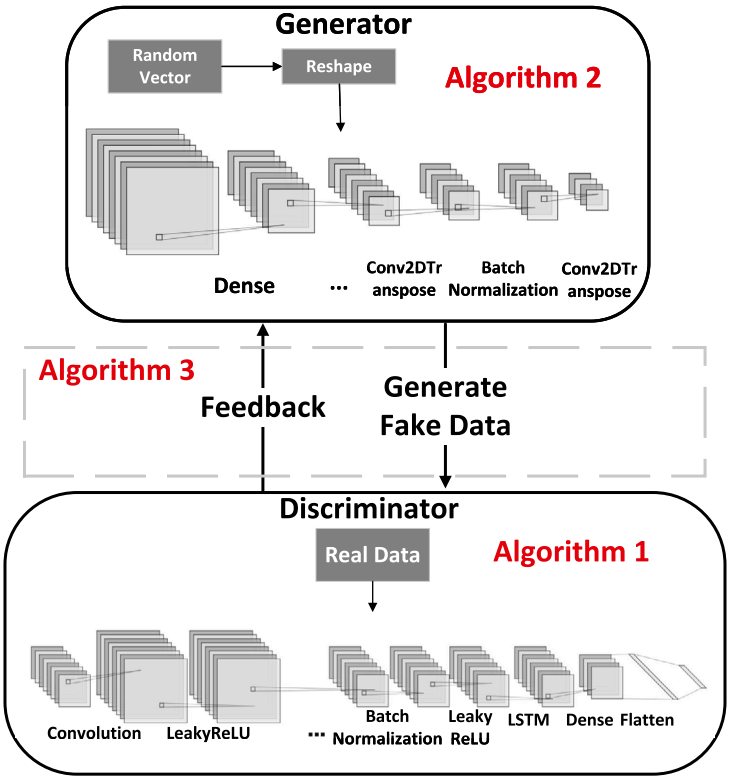

Fig. 7. The structure of CL-GAN.

real/fake data, and the maxpool layer can also be applied to control the convergence of the neural network (for example, to avoid over-fitting). Next, the LSTM of deep learning is applied to train data. The output of LSTM is sent to dropout layer to prevent overfitting. Subsequently, the last dense layer is employed to predict results. Finally, the Algorithm 3 shows that the discrimination result feeds back to the generator, enabling the generator to adjust its parameters based on the feedback and initiate a new round of generating fake data.

Algorithm 3 Define-GAN

Input: gen_model, d_model
Output: gan_model
1: $gan\_ouput \leftarrow d\_model(gen\_model.ouput)$
2: $gan\_model \leftarrow model(gen\_model.input, gan\_output)$
3: Return gan_model

The CL-GAN model utilizes CNN to extract and analyze the features of input traffic data. Then it combines LSTM with GAN to establish a normal behavior pattern, which can judge whether the input data are malicious traffic. The convolution layer and maxpool layer of CNN will be adjusted according to the input dataset. The LSTM will firstly process the input data L_n into LSTM layer, then process it in dense layer, dropout and flatten layers. Finally, sigmoid function is applied for binary classification. The sigmoid function is denoted by

$$f(j) = \frac{1}{1 + e^{-j}} \tag{10}$$

Sigmoid function is a binary classification function, which is more suitable for the HDA-IDS. The output of the neural network is compressed to (0,1) after its conversion, so the data is not easy to diverge in the process of transmission. The sigmoid function has the following characteristics: when j approaches negative infinity, $f(j)$ approaches 0; When j approaches positive infinity, $f(j)$ approaches 1; When $j = 0, f(j) = 1/2$.

Table 2
Hyperparameters selection.

Hyperparameters	NSL-KDD	CICIDS2018	Bot-IoT
Epoch	20	20	20
Batch size	10 000	10 000	100 000
CNN filter size	64	64	64
LSTM units	16	16	16
Dropout	0.3	0.3	0.3
Learning rate	0.0002	0.0002	0.0002

Table 3
The output size of each layer on CICIDS2018.

Layer	Output size	BN	Dropout	Activation
Input	$20 * 1$	✓	✓	LeakyReLu
Convolution	$20 * 64$	✓	✓	LeakyReLu
Convolution	$20 * 32$	✓	✓	LeakyReLu
Convolution	$20 * 32$	✓	✓	LeakyReLu
Convolution	$20 * 32$	✓	✓	LeakyReLu
LSTM	$20 * 16$			
Dense	$20 * 8$		✓	
Flatten	160		✓	
Dense	2			Sigmoid

4. Performance evaluation

4.1. Experiment setup

The hardware environment applied in this experiment is as follows: Intel(R) Xeon(R) CPU E5-2680 v4, GeForce RTX 3090 (24G VRAM) and 30G RAM. The software environment applied is as follows: Python 3.7.9, Jupyter Notebook, TensorFlow 2.5.0. In addition to these basic environments, the experiment also utilizes typical third-party libraries of python, such as pandas, keras, numpy, sklearn, etc. The hyperparameters selection of this experiment is shown in Table 2. Tables 3–5 lists the output size of each layer on three datasets.

Table 4
The output size of each layer on NSL-KDD.

Layer	Output size	BN	Dropout	Activation
Input	23 * 1	✓	✓	LeakyReLu
Convolution	23 * 64	✓	✓	LeakyReLu
Convolution	23 * 64	✓	✓	LeakyReLu
Convolution	23 * 32	✓	✓	LeakyReLu
Convolution	23 * 16	✓	✓	LeakyReLu
LSTM	23 * 16			
Dense	23 * 8		✓	
Flatten	184		✓	
Dense	2			Sigmoid

Table 5
The output size of each layer on Bot-IoT.

Layer	Output size	BN	Dropout	Activation
Input	10 * 1	✓	✓	LeakyReLu
Convolution	10 * 64	✓	✓	LeakyReLu
Convolution	10 * 64	✓	✓	LeakyReLu
Convolution	10 * 32	✓	✓	LeakyReLu
Convolution	10 * 16	✓	✓	LeakyReLu
Convolution	10 * 16	✓	✓	LeakyReLu
LSTM	10 * 16			
Dense	10 * 8		✓	
Flatten	80		✓	
Dense	2			Sigmoid

Table 6
Features selection.

Datasets	Features
CICIDS2018	Fwd IAT Max, Init Fwd Win Byts, Flow IAT Max, Fwd IAT Tot, Fwd Pkts/s, Flow Duration, Fwd IAT Mean, Flow IAT Mean, Fwd Header Len, TotLen Fwd Pkts, Subflow Fwd Byts, Fwd Seg Size Avg, Fwd Pkt Len Mean, Fwd Pkt Len Max, Pkt Size Avg, Pkt Len Mean, Flow IAT Std, Pkt Len Max, Pkt Len Std, Pkt Len Var
NSL-KDD	src_bytes, diff_srv_rate, same_srv_rate, count, dst_host_diff_srv_rate, dst_host_same_srv_rate, dst_host_srv_count, dst_host_srv_serror_rate, dst_bytes, dst_host_serror_rate, serror_rate, srv_serror_rate, logged_in, dst_host_same_src_port_rate, dst_host_count, srv_count, dst_host_srv_diff_host_rate, success_pred, srv_diff_host_rate, protocol_type_icmp, attack_type, land, dst_host_rerror_rate
Bot-IoT	srate, mean, max, stddev, min, N_IN_Conn_P_DstIP, state_number, N_IN_Conn_P_SrcIP, proto_tcp, proto_udp

4.2. Datasets

This experiment adopts the network layer public dataset CICIDS2018, NSL-KDD and the IoT dataset Bot-IoT. Aiming at the detection of DoS and botnet attacks, data related to these attacks are selected in this experiment. During this experiment, 80% of the dataset is allocated as the training set, followed by 10% as the validation set, and the final 10% of dataset is reserved for the testing set. Table 6 lists features with higher importance selected by FCBF calculation on three datasets.

4.2.1. CICIDS2018

The CICIDS2018 (Sharafaldin et al., 2018) applies the concept of profile to generate datasets in a systematic way. It includes detailed description and abstract distribution models of intrusion applications, protocols or lower-level network entities. It includes seven different attack scenarios: violent attack, heart bleeding, botnet, DoS, DDoS, Web attack and infiltration inside the network. There are 420 machines, 30 servers set as the victim organization, and 50 machines set as attack infrastructures. The CICFlowMeter-V3 extracts 80 features from the captured traffic. In this experiment, botnet, DoS and DDoS attack scenarios are selected for experiment. The specific selection is shown in Fig. 8.

4.2.2. NSL-KDD

The NSL-KDD dataset (Tavallaee et al., 2009) is selected because of its high utilization rate, and it is a traditional network traffic dataset. At the same time, selecting the NSL-KDD dataset is convenient for the subsequent evaluation between HDA-IDS and other systems. The NSL-KDD dataset is a revised version of the famous KDD99 dataset. It contains 43 features, which include 41 itself features and label. It has 4 different types of attacks: DoS, Probe, User-to-Root (U2R) and Remote-to-Local (R2L). In this experiment, DoS attacks and normal traffic are selected for subsequent processing. The specific selection is shown in Fig. 9.

4.2.3. Bot-IoT

The Bot-IoT dataset (Koroniotis et al., 2019) is chosen because it represents the environment where the IoT is attacked by DoS and botnet. It includes normal IoT network traffic and many kinds of attacks, which can be applied to evaluate the system proposed in this paper. There are 5 types of attacks, namely DDoS, DoS, OS/Service Scan, Keylogging and Data exfiltration attacks. Therefore, this experiment selects all the data of Bot-IoT to evaluate the HDA-IDS. The specific selection is shown in Fig. 10.

4.3. Evaluation metrics

The evaluation metrics of this experiment mainly involve confusion matrix (Fig. 11), accuracy, precision, recall and F1-score.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{11}$$

$$Precision = \frac{TP}{TP + FP} \tag{12}$$

$$Recall = \frac{TP}{TP + FN} \tag{13}$$

$$F1 - Score = \frac{2 \cdot \text{ Precision } \cdot \text{ Recall}}{\text{Precision } + \text{ Recall}} \tag{14}$$

True Positive (TP): TP indicates the number of normal traffic samples that are correctly classified as normal traffic.

False Negative (FN): FN indicates the number of normal traffic samples that are misclassified as attack traffic.

False Positive (FP): FP indicates the number of attack traffic samples that are misclassified as normal traffic.

True Negative (TN): TN indicates the number of attack traffic samples that are correctly classified as attack traffic.

4.4. HDA-IDS experimental results

The HDA-IDS is tested on CICIDS2018, NSL-KDD and Bot-IoT datasets. The detection and classification results of known attacks on CICIDS2018, NSL-KDD and Bot-IoT are shown in Fig. 12. The outcomes indicate that the application of Stacking technology employed in HDA-IDS surpasses that of AdaBoost, XGBoost, RF, GBDT in accuracy, precision, recall and F1-Score. This is mainly due to integrating the results of five basic model algorithms by stacking technology. Moreover, Hyperopt-TPE is applied to adjust the parameters to achieve the best experimental results.

Comparisons of known attacks on CICIDS2018, NSL-KDD and Bot-IoT are shown in Tables 7, 9, and 11. For known attacks, the stacking technology in HDA-IDS is superior to LSTM (Di Mauro et al., 2020), RNN (Hai & Nam, 2021), RF (Hai & Nam, 2021), Ensemble (Ludwig, 2017) in accuracy, precision, recall and F1-Score. Instead of organizing basic model results, the stacking of HDA-IDS organizes five basic models to train a final model. On one hand, stacking is capable of automatically blending the strengths of different models, thereby enhancing the overall performance of the model. On the other hand, the stacking process can be performed through cross-validation, effectively mitigating the issue of overfitting. Although the precision

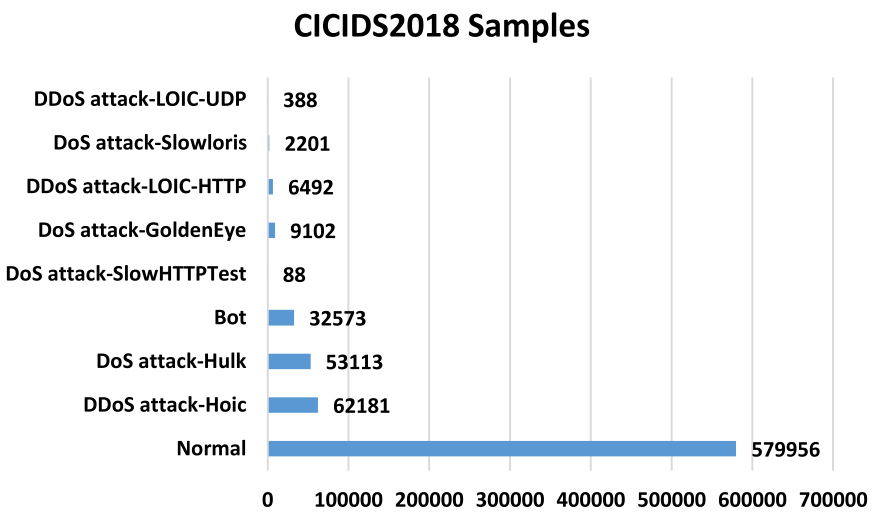

Fig. 8. CICIDS2018 samples related to IoT network.

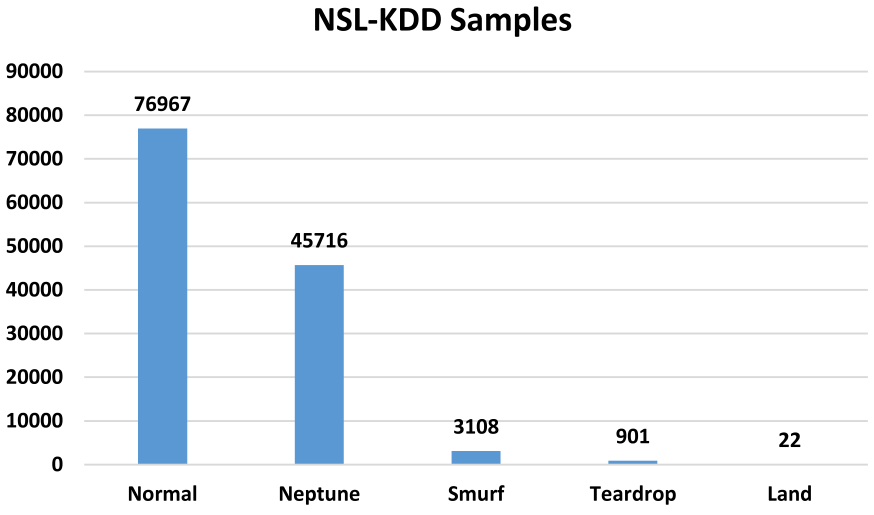

Fig. 9. NSL-KDD samples related to IoT network.

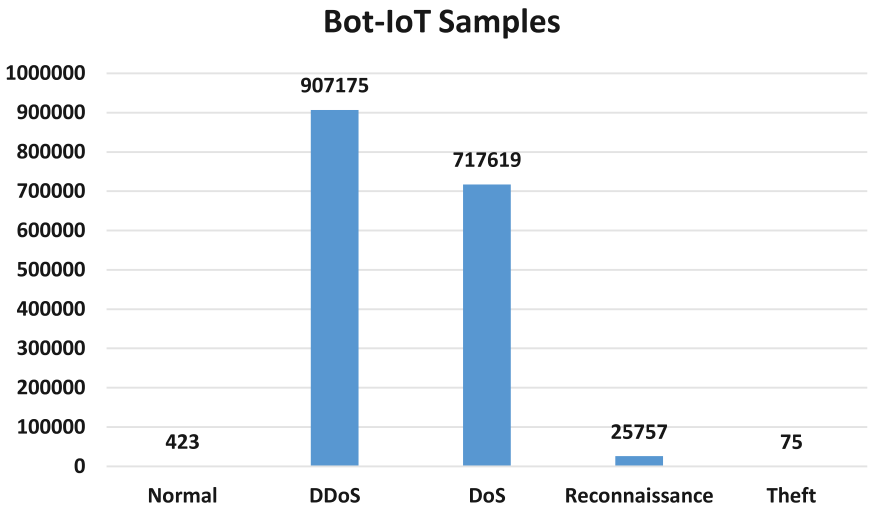

Fig. 10. Bot-IoT samples related to IoT network.

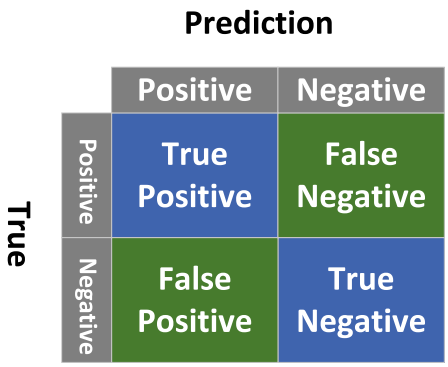

Fig. 11. Confusion matrix.

Table 7
Comparisons of known attacks detection on the CICIDS2018 dataset.

Method	Accuracy	Precision	Recall	F1-Score
SS-DEEP-ID (Abdel-Basset, Hawash, Chakrabortty, & Ryan, 2021)	98.71%	94.91%	94.30%	94.92%
LSTM (Di Mauro, Galatro, & Liotta, 2020)	97.01%	87.07%	92.67%	89.78%
DBN+ANN (Gamage & Samarabandu, 2020)	96.08%	92.46%	92.48%	92.47%
RNN (Hai & Nam, 2021)	93.85%	99.71%	95.48%	97.55%
RF (Hai & Nam, 2021)	95.07%	83.81%	86.21%	84.99%
Stacking	**99.95%**	**99.95%**	**99.95%**	**99.95%**

Table 8
Comparisons of unknown attacks detection on the CICIDS2018 dataset.

Method	Accuracy	Precision	Recall	F1-Score
SVM (Lin & Wang, 2002)	97.02%	97.05%	97.32%	97.18%
LSTM (Di Mauro et al., 2020)	98.01%	97.54%	96.5%	97.01%
AE+ANN (Gamage & Samarabandu, 2020)	98.22%	97.5%	98.22%	97.85%
ANN (Gamage & Samarabandu, 2020)	98.38%	98.54%	98.36%	98.44%
CL-GAN	**98.75%**	**98.81%**	**98.85%**	**98.82%**

Table 9
Comparisons of known attacks detection on the NSK-KDD dataset.

Method	Accuracy	Precision	Recall	F1-Score
S-NDAE (Shone, Ngoc, Phai, & Shi, 2018)	97.85%	100%	85.42%	87.37%
Ensemble (Ludwig, 2017)	92.49%	93%	92%	92%
DL (Diro & Chilamkurti, 2018)	99.20%	99.02%	99.27%	99.14%
Stacking	**99.98%**	**99.98%**	**99.98%**	**99.98%**

of S-NDAE (Shone et al., 2018), RNN (Koroniotis et al., 2019) and LSTM (Koroniotis et al., 2019) is slightly higher than the Stacking, the accuracy and recall of these methods are lower than the Stacking.

Table 10
Comparisons of unknown attacks detection on the NSK-KDD dataset.

Method	Accuracy	Precision	Recall	F1-Score
RNN-IDS (Yin et al., 2017)	81.29%	–	97.09%	–
AE (Ieracitano, Adeel, Morabito, & Hussain, 2019)	84.21%	87%	80.37%	81.98%
Q-SVM (Ieracitano et al., 2019)	83.15%	86.09%	79.86%	81.39%
ODM-ADS (Moustafa et al., 2019)	98.59%	–	98.25%	–
CL-GAN	**99.63%**	**99.63%**	**99.71%**	**99.66%**

Table 11
Comparisons of known attacks detection on the Bot-IoT dataset.

Method	Accuracy	Precision	Recall	F1-Score
Improved-IDS (Majhi et al., 2022)	99.33%	97.36%	–	98.34%
RNN (Koroniotis et al., 2019)	99.74%	99.99%	99.74%	–
LSTM (Koroniotis et al., 2019)	99.74%	99.99%	99.75%	–
FNN (Ge et al., 2019)	96.19%	98.9%	99.8%	99.4%
RNN and BiLSTM (Syed, Ge, & Baig, 2023)	99.55%	99.99%	99.02%	99.49%
Stacking	**99.92%**	**99.92%**	**99.92%**	**99.92%**

Table 12
Comparisons of unknown attacks detection on the Bot-IoT dataset.

Method	Accuracy	Precision	Recall	F1-Score
NB (Khanday et al., 2023)	88%	100%	77%	87%
LSTM (Khanday et al., 2023)	98%	98%	100%	99%
Linear SVC (Khanday et al., 2023)	88%	100%	76%	86%
CL-GAN	**98.53%**	**99.08%**	**98.53%**	**98.39%**

Table 13
Comparisons of semi-supervised models detection performance.

Semi-supervised method	Accuracy	Precision	Recall	F1-Score	Training time (min)
NSL-KDD					
AE+ANN (Gamage & Samarabandu, 2020)	76.9%	79.7%	76.93%	73.39%	0.5
DBN+ANN (Gamage & Samarabandu, 2020)	76.8%	78.94%	76.86%	73.36%	1.2
CL-GAN	**99.63%**	**99.63%**	**99.71%**	**99.66%**	**0.4**
CICIDS2018					
AE+ANN (Gamage & Samarabandu, 2020)	98.15%	97.34%	98.15%	97.63%	7.6
DBN+ANN (Gamage & Samarabandu, 2020)	98.33%	97.82%	98.33%	97.82%	10.5
CL-GAN	**98.75%**	**98.81%**	**98.85%**	**98.82%**	**6.3**

Comparisons of unknown attacks on CICIDS2018, NSL-KDD and Bot-IoT are shown in Tables 8, 10, and 12. For unknown attacks, the CL-GAN also performs well, and it is superior to AE + ANN (Gamage

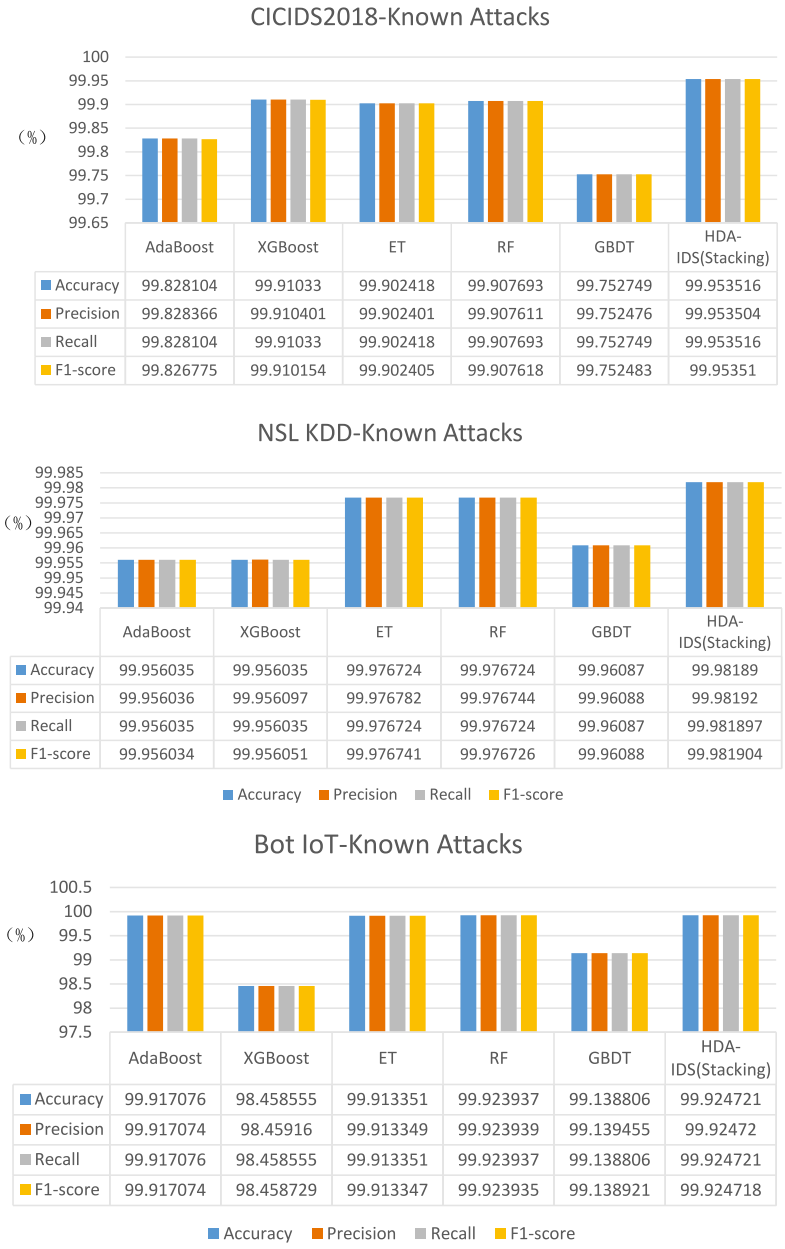

Fig. 12. Comparison stacking with AdaBoost, XGBoost, RF and GBDT on known attacks.

& Samarabandu, 2020), LSTM (Di Mauro et al., 2020), NB (Khanday et al., 2023), LSTM (Khanday et al., 2023) and Linear SVC (Khanday et al., 2023). To distinguish malicious traffic, the CL-GAN combines the CNN-LSTM with GAN to establish a normal behavior pattern. Compared with other neural networks, the CL-GAN has a feedback mechanism to generate high-quality data samples and make up for shortcomings in datasets. In addition, the CL-GAN distinguishes whether the generated samples are similar to real samples, and it is helpful to identify false data in traffic.

In comparison to other semi-supervised models, Table 13 presents the detection performance comparison of AE + ANN (Gamage & Samarabandu, 2020), DBN + ANN (Gamage & Samarabandu, 2020), and CL-GAN. It is evident that while AE + ANN (Gamage & Samarabandu, 2020) and DBN + ANN (Gamage & Samarabandu, 2020) exhibit similar accuracy rates to CL-GAN on the CICIDS2018 dataset, their performance is significantly inferior to CL-GAN on the NSL-KDD dataset. Additionally, the training time of AE + ANN (Gamage & Samarabandu, 2020) and DBN + ANN (Gamage & Samarabandu, 2020) exceeds that of CL-GAN. This notable difference can be attributed to CL-GAN's ability, as illustrated in Algorithm 2, to automatically adjust the number of CNN layers based on the dataset size. This crucial feature greatly reduces unnecessary execution time and prevents overfitting. Thus, CL-GAN stands out as a superior and efficient model for effectively detecting DoS and botnet attacks in the context of IoT.

Since the evaluation metrics applied in comparison papers are different from those in this paper, a small quantity of data in Tables 10 and 11 is missing. Meanwhile, the training loss and accuracy of CL-GAN on three datasets can be seen in Figs. 16, 17, and 18. From these figures, the training loss and accuracy tends to be stable on epoch 10, so this experiment trains it on 20 epochs.

The confusion metrics illustrating the detection of unknown attacks on three datasets are depicted in Figs. 13, 14, and 15. The effectiveness of CL-GAN in detecting unknown attacks is evident when examining these matrices. By discerning patterns of behavior distinct

Table 14
Comparisons of models testing time on NSL-KDD.

Models	LSTM (Gamage & Samarabandu, 2020)	RF (Gamage & Samarabandu, 2020)	AE+ANN (Gamage & Samarabandu, 2020)	DBN+ANN (Gamage & Samarabandu, 2020)	HDA-IDS
Testing time (s)	15.0	8.2	2.0	1.4	1.2

Table 15
Comparisons of models testing time on CICIDS2018.

Models	LSTM (Gamage & Samarabandu, 2020)	RF (Gamage & Samarabandu, 2020)	AE+ANN (Gamage & Samarabandu, 2020)	DBN+ANN (Gamage & Samarabandu, 2020)	HDA-IDS
Testing time (s)	24.0	25.5	1.9	1.3	1.0

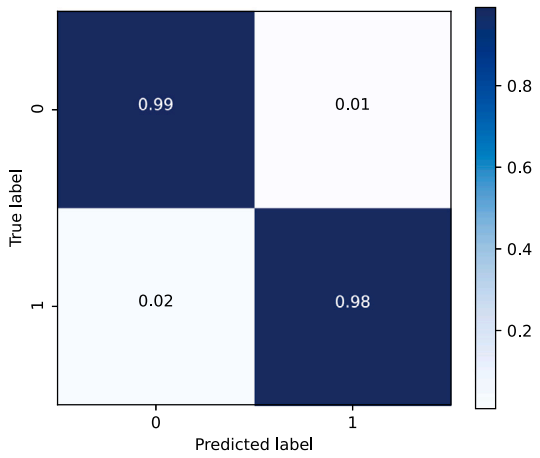

Fig. 13. Confusion matrix for detecting unknown attacks on the CICIDS2018 dataset.

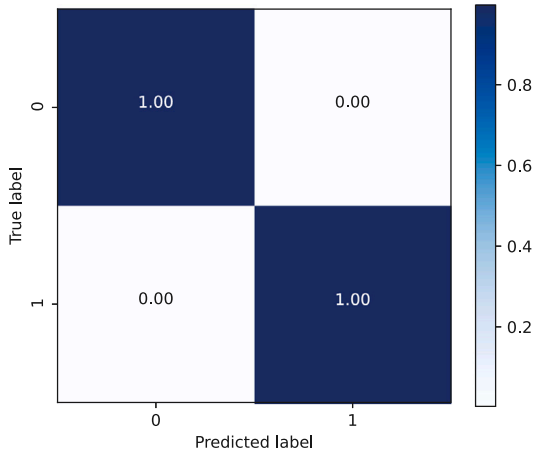

Fig. 14. Confusion matrix for detecting unknown attacks on the NSL-KDD dataset.

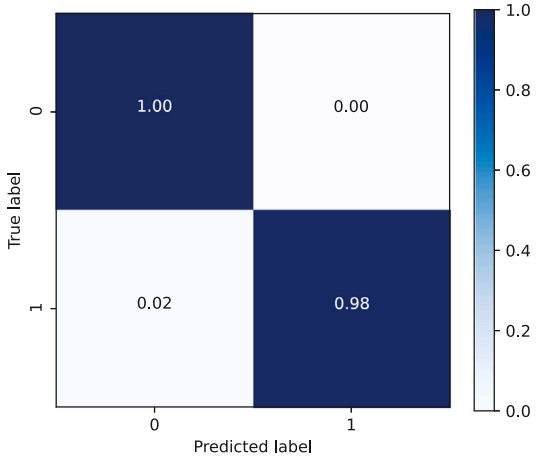

Fig. 15. Confusion matrix for detecting unknown attacks on the Bot-IoT dataset.

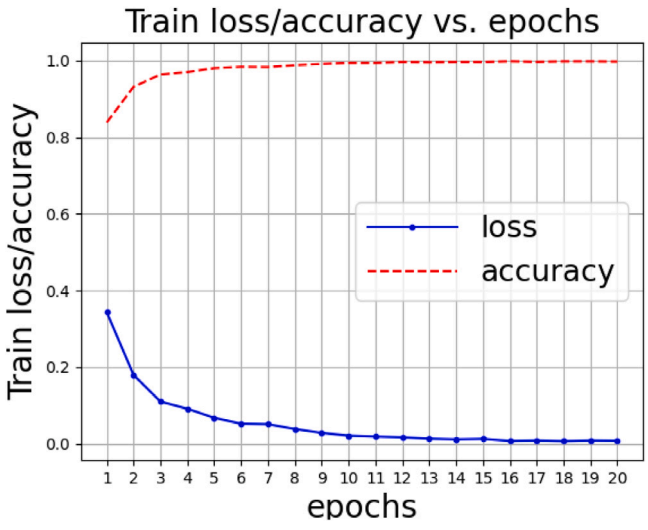

Fig. 16. Training loss and accuracy of CICIDS2018 dataset.

from established normal patterns, CL-GAN showcases its capability in accurately identifying unknown attacks. For smaller datasets like NSL-KDD, CL-GAN achieves nearly 100% detection accuracy. Moreover, for larger datasets such as CICIDS2018 and Bot-IoT, it attains a detection accuracy of 98%. The training loss and accuracy results are presented in Figs. 16, 17, and 18 for the three datasets. It can be observed that the training loss and accuracy stabilize at epoch 10, leading us to employ 20 epochs for training. Additionally, Table 16 outlines the loss metrics specifically for CL-GAN. The findings reveal that the CL-GAN model exhibits exceptional performance with consistently low loss values.

As demonstrated in Tables 14 and 15, both supervised model RF (Gamage & Samarabandu, 2020) and unsupervised model LSTM (Gamage & Samarabandu, 2020) exhibit the longest testing time, significantly surpassing the other three semi-supervised models (AE + ANN Gamage & Samarabandu, 2020, DBN + ANN Gamage & Samarabandu, 2020, and HDA-IDS). Among the remaining three semi-supervised models, our proposed HDA-IDS achieves the shortest testing time, reaching 1.2 s on the NSL-KDD dataset and 1.0 s on the CICIDS2018 dataset. Due to the limited prior research on testing and training time on the Bot-IoT dataset, information regarding these factors remains elusive.

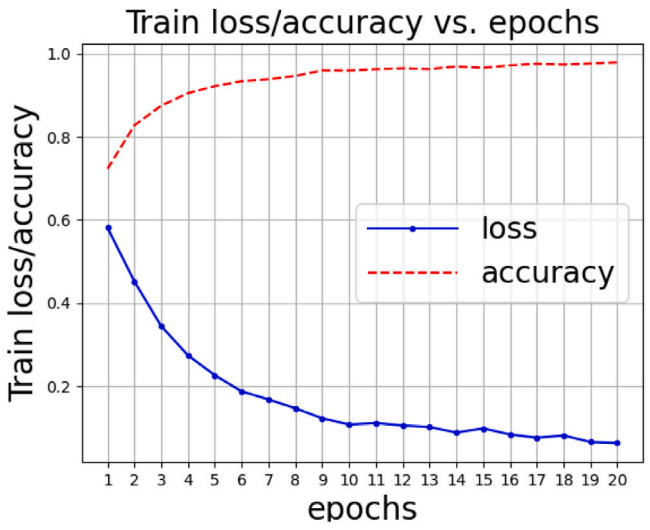

Fig. 17. Training loss and accuracy of NSL-KDD dataset.

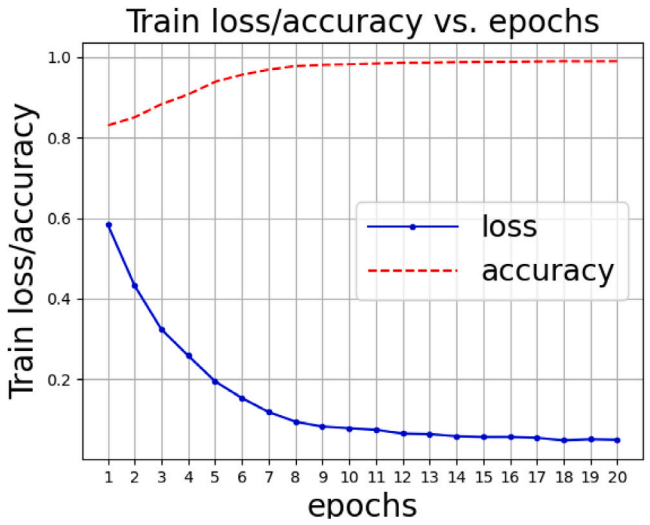

Fig. 18. Training loss and accuracy of Bot-IoT dataset.

Table 16
The loss of CL-GAN.

CL-GAN	NSL-KDD	CICIDS2018	Bot-IoT
GAN_{loss}	0.0128	0.1581	0.0576

5. Conclusion

This paper introduced a hybrid IDS denoted as HDA-IDS, encompassing both signature-based and anomaly-based approaches. For signature detection within HDA-IDS, a stacking technique was employed, wherein the outputs from XGBoost, GBDT, AdaBoost, RF, and ET served as inputs for classifying known instances of DoS and botnet attacks. In the realm of anomaly detection, the anomaly-based facet of HDA-IDS integrated the attributes of CNN-LSTM and GAN in the CL-GAN architecture. Notably, the CL-GAN exhibited robust performance in identifying unknown/zero-day attacks and effectively handling datasets characterized by imbalanced data distributions. Demonstrating swift detection times and minimal loss, the CL-GAN achieved accuracy rates of 99.63% on NSL-KDD, 98.53% on Bot-IoT, and 98.75% on CICIDS2018 datasets.

Benefiting from the utilization of stacking and CL-GAN, the HDA-IDS demonstrates dual capabilities. On one hand, it efficiently detects known DoS and botnet attacks. On the other hand, it facilitates data augmentation to address imbalanced datasets, while also employing a semi-supervised model to discern unknown instances of DoS and botnet attacks. Despite its advance of performance over existing literature, it still requires further efforts in improvement of detection range: the HDA-IDS primarily focuses on detecting the significant threats of DoS and botnet attacks prevalent in the IoT domain. However, our future work is to expand the detection capabilities of HDA-IDS to encompass a wider range of attacks commonly observed in IoT networks, including Man-in-the-Middle (MitM) attacks, eavesdropping, packet dropping, and other attacks.

CRediT authorship contribution statement

Sifan Li: Implements the idea of HDA-IDS. **Yue Cao:** Supervisor to response this work. **Shuohan Liu:** Experiment dataset. **Yuping Lai:** Designed the detection modeling of HAD-IDS. **Yongdong Zhu:** Summarize literature, Shape application of HDA-IDS. **Naveed Ahmad:** Shape the language, Organization of paper.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

Hubei Province Key Research and Development Program (2021BAA027) and Wuhan AI Innovation Program (2023010402040020) and Fundamental Research Funds for the Central Universities, China (2042022rc0020).

References

Abdel-Basset, M., Hawash, H., Chakrabortty, R. K., & Ryan, M. J. (2021). Semi-supervised spatiotemporal deep learning for intrusions detection in IoT networks. *IEEE Internet of Things Journal*, *8*(15), 12251–12265.

Abdelmoumin, G., Rawat, D. B., & Rahman, A. (2021). On the performance of machine learning models for anomaly-based intelligent intrusion detection systems for the internet of things. *IEEE Internet of Things Journal*, *9*(6), 4280–4290.

Alaba, A., Maitanmi, S., & Ajayi, O. (2019). An ensemble of classification techniques for intrusion detection systems. *International Journal of Computer Science and Information Security (IJCSIS)*, *17*(11).

Alferaidi, A., Yadav, K., Alharbi, Y., Razmjooy, N., Viriyasitavat, W., Gulati, K., et al. (2022). Distributed deep CNN-LSTM model for intrusion detection method in IoT-based vehicles. *Mathematical Problems in Engineering*, *2022*.

Ashraf, J., Keshk, M., Moustafa, N., Abdel-Basset, M., Khurshid, H., Bakhshi, A. D., et al. (2021). IoTBoT-IDS: A novel statistical learning-enabled botnet detection framework for protecting networks of smart cities. *Sustainable Cities and Society*, *72*, Article 103041.

Aswal, K., Dobhal, D. C., & Pathak, H. (2020). Comparative analysis of machine learning algorithms for identification of BOT attack on the Internet of Vehicles (IoV). In *2020 international conference on inventive computation technologies (ICICT)* (pp. 312–317). IEEE.

Aydın, M. A., Zaim, A. H., & Ceylan, K. G. (2009). A hybrid intrusion detection system design for computer network security. *Computers & Electrical Engineering*, *35*(3), 517–526.

Chauhan, R., & Shah Heydari, S. (2020). Polymorphic adversarial DDoS attack on IDS using GAN. In *2020 international symposium on networks, computers and communications (ISNCC)* (pp. 1–6). http://dx.doi.org/10.1109/ISNCC49221.2020.9297264.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, *16*, 321–357.

Chowdhury, R., Sen, S., Goswami, A., Purkait, S., & Saha, B. (2023). An implementation of bi-phase network intrusion detection system by using real-time traffic analysis. *Expert Systems with Applications*, *224*, Article 119831.

Dash, T. (2017). A study on intrusion detection using neural networks trained with evolutionary algorithms. *Soft Computing*, *21*(10), 2687–2700.

Di Mauro, M., Galatro, G., & Liotta, A. (2020). Experimental review of neural-based approaches for network intrusion management. *IEEE Transactions on Network and Service Management*, *17*(4), 2480–2495.

Diro, A. A., & Chilamkurti, N. (2018). Distributed attack detection scheme using deep learning approach for Internet of Things. *Future Generation Computer Systems*, *82*, 761–768.

Džeroski, S., & Ženko, B. (2004). Is combining classifiers with stacking better than selecting the best one? *Machine Learning*, *54*(3), 255–273.

Fei, N., Gao, Y., Lu, Z., & Xiang, T. (2021). Z-score normalization, hubness, and few-shot learning. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 142–151).

Ferdowsi, A., & Saad, W. (2019). Generative adversarial networks for distributed intrusion detection in the internet of things. In *2019 IEEE global communications conference (GLOBECOM)* (pp. 1–6). IEEE.

Gamage, S., & Samarabandu, J. (2020). Deep learning methods in network intrusion detection: A survey and an objective comparison. *Journal of Network and Computer Applications*, *169*, Article 102767.

Garrido-Merchán, E. C., & Hernández-Lobato, D. (2020). Dealing with categorical and integer-valued variables in bayesian optimization with gaussian processes. *Neurocomputing*, *380*, 20–35.

Ge, M., Fu, X., Syed, N., Baig, Z., Teo, G., & Robles-Kelly, A. (2019). Deep learning-based intrusion detection for IoT networks. In *2019 IEEE 24th pacific rim international symposium on dependable computing (PRDC)* (pp. 256–25609). IEEE.

Hai, T. H., & Nam, L. H. (2021). A practical comparison of deep learning methods for network intrusion detection. In *2021 international conference on electrical, communication, and computer engineering (ICECCE)* (pp. 1–6). IEEE.

Hewamalage, H., Bergmeir, C., & Bandara, K. (2021). Recurrent neural networks for time series forecasting: Current status and future directions. *International Journal of Forecasting*, *37*(1), 388–427.

Hodo, E., Bellekens, X., Hamilton, A., Dubouilh, P.-L., Iorkyase, E., Tachtatzis, C., et al. (2016). Threat analysis of IoT networks using artificial neural network intrusion detection system. In *2016 international symposium on networks, computers and communications (ISNCC)* (pp. 1–6). IEEE.

Ieracitano, C., Adeel, A., Morabito, F. C., & Hussain, A. (2019). A novel statistical analysis and autoencoder driven intelligent intrusion detection approach. *Neurocomputing*, *387*.

Karami, A. (2018). An anomaly-based intrusion detection system in presence of benign outliers with visualization capabilities. *Expert Systems with Applications*, *108*, 36–60.

Khanday, S. A., Fatima, H., & Rakesh, N. (2023). Implementation of intrusion detection model for DDoS attacks in Lightweight IoT Networks. *Expert Systems with Applications*, *215*, Article 119330.

Khraisat, A., Gondal, I., Vamplew, P., Kamruzzaman, J., & Alazab, A. (2019). A novel ensemble of hybrid intrusion detection system for detecting internet of things attacks. *Electronics*, *8*(11), 1210.

Koroniotis, N., Moustafa, N., Sitnikova, E., & Turnbull, B. (2019). Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset. *Future Generation Computer Systems*, *100*, 779–796.

Lampe, B., & Meng, W. (2023). A survey of deep learning-based intrusion detection in automotive applications. *Expert Systems with Applications*, *221*, Article 119771.

Li, T., Zhang, Y., & Wang, T. (2021). SRPM–CNN: a combined model based on slide relative position matrix and CNN for time series classification. *Complex & Intelligent Systems*, *7*(3), 1619–1631.

Lima, L. L., Ferreira Junior, J. R., & Oliveira, M. C. (2021). Toward classifying small lung nodules with hyperparameter optimization of convolutional neural networks. *Computational Intelligence*, *37*(4), 1599–1618.

Lin, C.-F., & Wang, S.-D. (2002). Fuzzy support vector machines. *IEEE Transactions on Neural Networks*, *13*(2), 464–471.

Liu, Y., Zhi, T., Shen, M., Wang, L., Li, Y., & Wan, M. (2022). Software-defined DDoS detection with information entropy analysis and optimized deep learning. *Future Generation Computer Systems*, *129*, 99–114.

Ludwig, S. A. (2017). Intrusion detection of multiple attack classes using a deep neural net ensemble. In *2017 IEEE symposium series on computational intelligence (SSCI)* (pp. 1–7). IEEE.

MacQueen, J. B. (1965). *On the asymptotic behavior of k-means*: *Technical Report*, CALIFORNIA UNIV LOS ANGELES WESTERN MANAGEMENT SCIENCE INST.

Majhi, B., et al. (2022). An improved intrusion dectection system using BoT-IoT dataset. In *2022 IEEE 11th international conference on communication systems and network technologies (CSNT)* (pp. 488–492). IEEE.

Moustafa, N., Choo, K.-K. R., Radwan, I., & Camtepe, S. (2019). Outlier dirichlet mixture mechanism: Adversarial statistical learning for anomaly detection in the fog. *IEEE Transactions on Information Forensics and Security*, *14*(8), 1975–1987.

Nguyen, X.-H., & Le, K.-H. (2023). Robust detection of unknown DoS/DDoS attacks in IoT networks using a hybrid learning model. *Internet of Things*, *23*, Article 100851.

Pan, S., Morris, T., & Adhikari, U. (2015). Developing a hybrid intrusion detection system using data mining for power systems. *IEEE Transactions on Smart Grid*, *6*(6), 3104–3113. http://dx.doi.org/10.1109/TSG.2015.2409775.

Rahman, M. A., Asyhari, A. T., Leong, L., Satrya, G., Hai Tao, M., & Zolkipli, M. (2020). Scalable machine learning-based intrusion detection system for IoT-enabled smart cities. *Sustainable Cities and Society*, *61*, Article 102324.

Rodríguez, P., Bautista, M. A., Gonzalez, J., & Escalera, S. (2018). Beyond one-hot encoding: Lower dimensional target embedding. *Image and Vision Computing*, *75*, 21–31.

Rosay, A., Carlier, F., & Leroux, P. (2020). Feed-forward neural network for network intrusion detection. In *2020 IEEE 91st vehicular technology conference (VTC2020-Spring)* (pp. 1–6). IEEE.

Sarjan, H., Ameli, A., & Ghafouri, M. (2022). Cyber-security of industrial internet of things in electric power systems. *IEEE Access*, *10*, 92390–92409. http://dx.doi.org/10.1109/ACCESS.2022.3202914.

Seo, E., Song, H. M., & Kim, H. K. (2018). GIDS: GAN based intrusion detection system for in-vehicle network. In *2018 16th annual conference on privacy, security and trust (PST)* (pp. 1–6). http://dx.doi.org/10.1109/PST.2018.8514157.

Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. In *ICISSp, Vol. 1* (pp. 108–116).

Shone, N., Ngoc, T. N., Phai, V. D., & Shi, Q. (2018). A deep learning approach to network intrusion detection. *IEEE Transactions on Emerging Topics in Computational Intelligence*, *2*(1), 41–50.

Shunmugapriya, P., & Kanmani, S. (2013). Optimization of stacking ensemble configurations through artificial bee colony algorithm. *Swarm and Evolutionary Computation*, *12*, 24–32.

Syed, N. F., Ge, M., & Baig, Z. (2023). Fog-cloud based intrusion detection system using Recurrent Neural Networks and feature selection for IoT networks. *Computer Networks*, Article 109662.

Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009). A detailed analysis of the KDD CUP 99 data set. In *2009 IEEE symposium on computational intelligence for security and defense applications* (pp. 1–6). http://dx.doi.org/10.1109/CISDA.2009.5356528.

Vadursi, M., Ceccarelli, A., Duarte, E. P., & Mahanti, A. (2016). System and network security: anomaly detection and monitoring. *Journal of Electrical and Computer Engineering*, *2016*.

Vishwakarma, R., & Jain, A. K. (2020). A survey of DDoS attacking techniques and defence mechanisms in the IoT network. *Telecommunication Systems*, *73*(1), 3–25.

Yang, L., Moubayed, A., & Shami, A. (2021). MTH-IDS: a multitiered hybrid intrusion detection system for Internet of vehicles. *IEEE Internet of Things Journal*, *9*(1), 616–632.

Yang, L., & Shami, A. (2020). On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing*, *415*, 295–316.

Yin, C., Zhu, Y., Fei, J., & He, X. (2017). A deep learning approach for intrusion detection using recurrent neural networks. *IEEE Access*, *5*, 21954–21961.

Yu, L., & Liu, H. (2003). Feature selection for high-dimensional data: A fast correlation-based filter solution. In *Proceedings of the 20th international conference on machine learning (ICML-03)* (pp. 856–863).