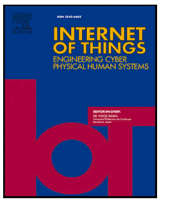
Internet of Things 26 (2024) 101212

Contents lists available at ScienceDirect

Internet of Things

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



Research article

SYN-GAN: A robust intrusion detection system using GAN-based synthetic data for IoT security

Saifur Rahman, Shantanu Pal *, Shubh Mittal, Tisha Chawla, Chandan Karmakar *

School of Information Technology, Deakin University, Melbourne, Victoria, Australia

ARTICLE INFO

Keywords:
GAN
NIDS
Synthetic data
Machine learning
Internet of Things

ABSTRACT

As technological communication progresses, diverse datasets are exchanged across distributed environments using the Internet of Things (IoT). However, the IoT environment is vulnerable to attacking and breaching data privacy or making a robust system worse by providing attack data. To address potential risks of attacks, researchers have been conducting experiments on network intrusion detection systems (NIDS) to mitigate threats effectively. The issue of data imbalance and associated data collection costs persists, hindering the ability of machine learning (ML) models to learn malicious behaviour effectively and consequently impacting the accuracy of network threat detection. Addressing these issues, our study explores the potential of using 100% synthetic data generated via Generative Adversarial Networks (GAN) for training ML models in Network Intrusion Detection Systems (NIDS). This approach reduces the dependency on real-world data significantly, paving the way for a more flexible and ethically convenient model-building process. For the UNSW-NB15 dataset, we achieved an accuracy of 90%, a precision of 91%, a recall of 90%, and an F1 score of 89%. For the NSL-KDD dataset, our results showed an accuracy of 84%, a precision of 85%, a recall of 84%, and an F1 score of 84%. For the BoT-IoT dataset, we attained perfect scores of 100% across all metrics. These outcomes indicate that the values obtained from our analysis demonstrate high performance, yielding comparative or superior results to previous studies. Therefore, our study successfully replicates real-world network intrusion detection data, showing new opportunities for the use of generative data in cyber security.

1. Introduction

The emergence of the Internet of Things (IoT), coupled with the high communication rates facilitated by fifth (5G) and sixth-generation (6G) technologies, has led to the advancing the remote data monitoring and automated decision-making systems. However, the access point for remote monitoring systems is vulnerable to potential threats, unauthorised access, data breaches, or other security concerns. [1].

Network Intrusion Detection Systems (NIDS) is essential as a defence mechanism for computer networks against potential threats like DoS, DDoS, theft, shellcode, and backdoors. These mechanisms monitor network traffic, identify potential threats, and protect against malicious activities. Traditional NIDS, such as signature-base, analyse network traffic for known attack signatures or patterns indicative of abnormal behaviour. Upon detecting malicious traffic, systems require promiscuous access to network data, triggering an alert. However, attack signatures or patterns indicative of abnormal behaviour method, while somewhat effective, often lag behind the rapidly evolving landscape of cyber threats [2], including advanced persistent threats (APTs) [3], zero-day exploits,

* Corresponding authors.
E-mail address: shantanu.pal@deakin.edu.au (S. Pal).

https://doi.org/10.1016/j.iot.2024.101212
Received 6 March 2024; Received in revised form 22 March 2024; Accepted 2 May 2024
Available online 7 May 2024

and polymorphic malware, which can evade detection. This distinction arises because traditional security approaches struggle to keep pace with the dynamic nature of cyber threats. As threat actors continually develop new tactics and exploit vulnerabilities, the time required to identify these risks, implement security updates, and adapt defensive strategies can create a significant gap. Moreover, the risk posed by these threats is compounded by the increasing volume of network traffic and the diversity of network protocols and devices, especially with the advent of the IoT and cloud computing. This evolving threat landscape necessitates a more dynamic approach to network security. The integration of Machine Learning (ML) into NIDS advancing and enhancing significantly improved network monitoring capabilities [4]. ML-based systems are adept at identifying malicious patterns in network traffic, which traditional methods often fail to detect [5]. These systems depend on training and necessitate a substantial amount of labelled data. This adaptability allows ML systems to detect previously unknown threats that might go undetected by static, rule-based security measures. The quality data is crucial, as it forms the foundation upon which an ML-based NIDS learns and adapts. However, the optimisation of ML-based NIDS faces a significant challenge due to the scarcity of high-quality of training data [6]. This limitation poses a substantial challenge in the training and evaluation of NIDS.

In recent years, there has been a paradigm shift towards the use of generative Models, particularly Generative Adversarial Networks (GANs) [7]. GANs consist of two neural networks: a *Generator* that creates data resembling real data from random noise and a *Discriminator* that learns to differentiate between real and generated data. Through iterative training, the Generator improves its data creation, while the Discriminator enhances its detection capabilities. This adversarial process enables GANs to produce high-quality synthetic data close to real-world data, useful in various applications in healthcare, where it has been used for X-ray security screening [8], artifact removal in medical imaging [9], emotion recognition using electroencephalogram (EEG) data [10], and deep fake face generation [11]. Beyond healthcare, GANs have also found significant applications in other domains. These use cases have paved the way for exploring GANs in cybersecurity for anomaly detection and class balancing in model training for Deep Learning (DL) models. However, to the best of our knowledge, a gap remains in the available literature, as no available study has yet focused on fully utilising synthetic data for model training in the context of NIDS instead of real-world data. The reason for choosing synthetic data instead of real-world data collection is that it is costly and difficult to capture all the attack types in a single data collection [12]. Through the fusion of GANs with anomaly detection, our novel method (more discussion on Section 3) illustrates that training solely on synthetic data is sufficient for identifying attacks, alleviating the need for extensive, real-world data collection. However, it is essential to clarify that a GAN is trained on a specific attack type to generate synthetic data during training. In the context of attack detection, the efficacy relies on the classifier, as outlined in Table 5. Therefore, when encountering new attack types, training the classifier with these new attack types is necessary. GANs are solely utilised to augment the dataset by leveraging reference attack data. It is important to note that GANs do not serve as classifiers themselves; rather, they are employed to mitigate the cost associated with collecting data for training classifiers with a large dataset to enhance the robustness of NIDS classifiers.

We provide a comprehensive framework of our system The framework comprises four distinct phases (refer to Fig. 1). From top to bottom they are: (1) data preparation; (2) GANs model for synthetic data generation; (3) training various ML models for attack detection; and (4) evaluating ML models using real-world data to determine the optimal model selection. In the data preparation step, the system eliminates noisy data and maintains the standardisation of features. After data preparation, the system trains the GANs model to generate synthetic data for different types of attacks. After generating synthetic data, we trained logistic regression, Decision tree, Random Forest, Gradient Boosting, AdaBoost, Support Vector Classifier, K-nearest neighbours and Gaussian Naive Bayes models for attack detection. Finally, evaluated real-world data from UNSW-NB15, NSL-KDD dataset and BoT-IoT datasets [13–15] are used to determine the optimal model selection. The major contributions of the paper can be summarised as follows:

- We propose a novel NIDS by integrating cutting-edge GANs to generate synthetic data and train various ML models on synthetic data.
- Through comparative experiments with various ML models tested with real-world data, we prove that training with synthetic data can detect attacks from real-world data from UNSW-NB15, NSL-KDD, and BoT-IoT sources.
- Our study shows that Gaussian Naive Bayes (GaussianNB) and K-Nearest Neighbours (KNN) performance are superior in the context of NIDS compared to other ML models.

The rest of this paper is structured as follows: Section 2 provides a concise overview of related studies in NIDS employing ML methodologies. Section 3 explains our methodology and the proposed framework, elaborating on the four main stages, such as data preparation, GANs model for synthetic data generation, training various ML models for attack detection and evaluating ML models using real-world data to determine the optimal model selection. In Section 4, we present experimental outcomes with meticulous analysis. Section 5 discusses the significant findings of the experiments. Finally, Section 6 offers concluding remarks and outlines future research directions.

2. Related work

In this section, we discuss the existing literature on the application of ML and DL techniques in NIDS, with a particular focus on how these technologies have been utilised for attack classification. Additionally, we explore the emerging role of GANs in this field. ML and DL-based attack classification techniques have been commonly used for anomaly detection, log analysis, heuristic-based detection, and traffic volume monitoring in NIDS. While ML algorithms like Artificial Neural Networks (ANN), Support Vector Machines (SVM), and Naive Bayes (NB) are shallow models used for supervised learning tasks in NIDS; DL algorithms like GAN, Autoencoders (AE), and Restricted Boltzmann Machines (RBM) are used for unsupervised learning tasks in NIDS [16]. For example,

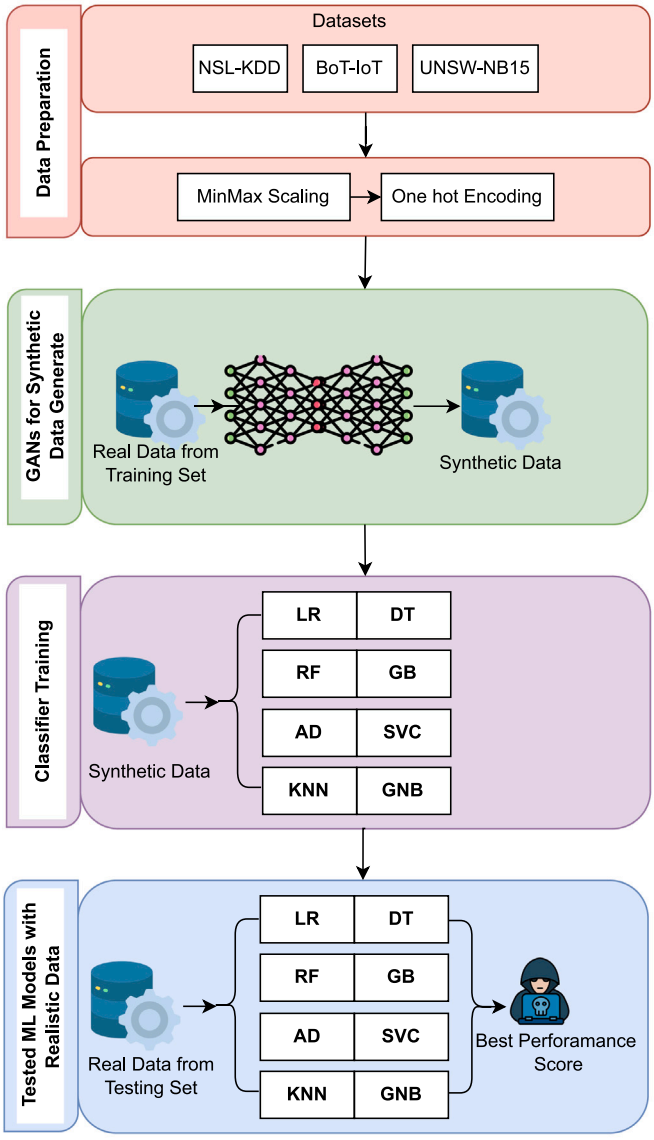


Fig. 1. Illustration of our proposed framework utilising GAN for generating synthetic training data and employing ML classifiers with real-time testing data for NIDS.

Dutta et al. [17] proposed a DL-ensembled approach that addresses the evolving challenges of network intrusion detection. By leveraging sophisticated DL techniques, the paper seeks to enhance the reliability and efficiency of detecting attacks in network security systems. However, unlike ours, their experiment did not solely depend on synthetic data, and real-world data collection still exists.

Almaraz-Rivera et al. [18] demonstrated the application of ML and DL models in detecting DDoS attacks on IoT devices, emphasising log analysis in identifying such threats. Similarly, Jiang et al. [19] discussed the integration of ML in additive manufacturing, which can be analogous to heuristic-based detection in NIDS, where ML algorithms learn from historical data to identify potential threats. While the above methods utilising traditional ML and DL algorithms have shown promising results with a high attack classification rate in various NIDS use cases, one common limitation is the real-world dependency on training data for model training, which is costly and involves ethical considerations. This shows there has been a recent shift towards generative models for mimicking real-time data for NIDS. Generative models, particularly GANs, have been used frequently in NIDS for data sampling, data creation, and anomaly detection. Studies have focused on employing GANs for synthetic data creation to balance minority classes, which are underrepresented in the dataset, thereby removing bias during model training. For instance, Kumar et al. [20] utilised XGBoost-WCGAN on datasets such as NSL-KDD, UNSW-NB15, and BoT-IoT, to demonstrate the effectiveness of GAN variants in class balancing for producing high detection rates. Similarly, Zhao et al. [21] addressed data imbalance issues in the

Table 1
Comparison of dependency on real training data and utilisation of latent space reduction in existing literature versus our proposed approach.

Reference	Real data	Synthetic data	Combined data	Datasets	Real data dependency	Classifier
[20]	✓	✓	✓	3	✓	XGBoost + WCGAN
[26]	✓	✓	✓	2	✓	Bi-GAN
[28]	✓	✓	✓	1	✓	RF + CTGAN, RF + TVAE
[21]	✓	✓	✓	1	✓	CTGAN
[23]	✓	✓	✓	1	✓	RF + GAN
[24]	✓	✓	✓	1	✓	GAN
[22]	✓	✓	✓	1	✓	RF + GAN
[25]	✓	✓	✓	1	✓	BiGAN
(Proposed)	✓	✓	✗	3	✗	GaussianNB + GAN, KNN + GAN

Table 2
List of acronyms and their full forms.

Acronym	Full form
DL	Deep learning
ML	Machine learning
NIDS	Intrusion detection system
GAN	Generative adversarial network
RF	Random forest
GB	Gradient boosting
AB	Adaboost
DT	Decision trees
SVC	Support vector classifier
LR	Logistic regression
GNB	Gaussian naive Bayes
KNN	K nearest neighbour

NSL-KDD dataset using conditional GANs, preventing missing data and low recognition rates of scarce data types by utilising real and synthetic data generated. Furthermore, Lee et al. [22] found that GAN combined with Random Forest (RF) was more effective than the Synthetic Minority Oversampling Technique (SMOTE) for class balancing, utilising the CICIDS 2017 dataset.

Further studies have focused on deploying classification models with GANs for a higher attack detection rate in different network settings. For example, Ding et al. [23] used a custom-built GAN model in predicting attacks in complex network landscapes, paving the way for more secure and resilient systems in the era of 5G-enabled metaverse environments. Seo et al. [24] applied GANs in in-vehicle network systems to develop an effective intrusion detection system tailored to the unique requirements of vehicle networks. Similarly, Chen et al. [25] highlighted the need for better speedup and synchronisation in training GAN models with the KDD-99 dataset using BiGAN, considering the sequence and timing of events. While Xu et al. [26] emphasised the need for detecting network intrusions without unnecessary training steps and with a simpler loss function, as seen in the NSL-KDD and IC-DDoS2019 datasets, allowing the generator and the discriminator to be trained without needing to be in sync in their training iterations. Nevertheless, in the domain of anomaly detection, where supervised learning often shows limitations, Kaplan et al. [27] suggested updating the generator independent from the discriminator to improve the model's performance. The above studies emphasise the need to improve GAN model stability and training efficiency, particularly in dynamic and fast-paced environments such as NIDS.

However, limited studies have been conducted on creating real-time data with GANs for training ML and DL models (cf. Table 1). For example, Chale et al. [28] focused on generating real-world cyber data for training ML classifiers. Using the Defense Research and Engineering Network (DREN) dataset and employing RF with CGAN and tabular variational autoencoder (TVAE), they created synthetic data statistically similar to real data. However, they found that classifiers trained solely on synthetic data underperformed, suggesting the inclusion of at least 15% real data in training setups. Their study is limited to the DREN data, and a significant amount of training data is needed to improve the performance metrics results. Therefore, Our study aims to train ML models using 100% synthetic data generated from GANs. To achieve this, we conduct experiments with three datasets—NSL-KDD, UNSW-NB15, and BoT-IoT—widely utilised in recent NIDS research, as detailed in Section 3.2.1.

As real-world data from cyber attacks is limited, our approach could save significant resources in generating real-time datasets reducing dependency on real data of model training. In Table 2, we list acronyms and their full forms.

3. Methodology

In this section, we discuss the use case scenarios, problem formulation, synthetic data generation process, ML model for NIDS, and evaluation metrics to justify the strength of our model.

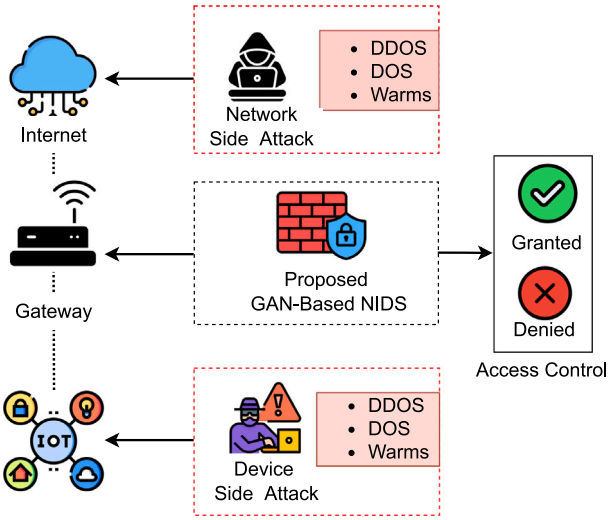


Fig. 2. A practical use case scenario of our proposed GAN-based NIDS.

3.1. A practical use case scenario

With the rise of IoT, numerous devices connect to networks. IoT facilitates remote control and maintenance via diverse network connections. Despite its convenience, IoT faces escalating attack risks [29]. As illustrated in Fig. 2, IoT devices and networks encounter various vulnerabilities. To counter these threats, our proposed GAN-driven NIDS model is installed in the gateway. A GAN-driven NIDS entails training a traditional ML model using synthetic data generated by GAN. There is a negligible delay and computationally expensive in real-world deployment because the trained machine learning model is installed without running the GAN model to generate synthetic data. This is because no training session is involved once the trained ML model has been deployed, and there is no significant computational complexity as GAN is not involved after training the ML model. Initially, it detects external network traffic to prevent remote attacks (Worms, Reconnaissance, DoS attacks, etc.). Subsequently, it monitors internal network traffic to block device entity invasions (brute force, unauthorised access, etc.) by hackers. Upon detecting malicious traffic, the model alerts the access control system, aiding prompt managerial intervention.

3.2. Data preparation

This section describes the preprocessing of the UNSW-NB15, NSL-KDD, and BoT-IoT datasets to prepare training and testing sets.

3.2.1. Dataset

UNSW-NB15 dataset: This dataset was developed by the Cyber Range Lab of the Australian Centre for Cyber Security (ACCS) [13] using the IXIA PerfectStorm tool for network intrusion detection research. The dataset contains 2,540,044 records, divided into a training set of 175,341 records and a testing set of 82,332 records. The dataset features 49 attributes, including various types of attack signatures and normal traffic features. It categories network traffic into ten classes, comprising nine types of attacks (Analysis, DoS, DoS, Backdoor, Probe, DDoS, DoS, R2L, Reconnaissance, Exploits, U2R, Theft, Fuzzers, Generic, Shellcode, Worms) and a normal class.

NSL-KDD dataset: This dataset [30] is an improved version of the KDD'99 dataset, designed to overcome the limitations of its preceding dataset [14]. NSL-KDD contains a total of 148,517 records, with 125,973 in the training set and 22,544 in the test set. The dataset features 41 attributes and categorises network traffic into four attack classes (DoS, Probe, R2L, U2R) and one normal class.

BoT-IoT dataset: This dataset was developed in the Cyber Range Lab of UNSW Canberra [15]. It was designed to simulate a realistic network environment that incorporates both normal and botnet traffic. It contains over 72 million records, with each record featuring 46 attributes. The dataset categorises network traffic into four attack classes (Dos, DDOS, Reconnaissance) and one normal class. This dataset is significant for its focus on IoT-specific security challenges such as Keylogging, Data Exfiltration, DoS and DDoS, offering a range of scenarios for training and testing NIDS models in IoT environments.

We have removed the outlier datasets, such as null data and non-numerical values in numerical attributes. To fair comparison, we have normalised all datasets using MinMax scaling defined by the following equation:

$$\widetilde{x} = \frac{x_i - min(x)}{max(x) - min(x)} \tag{1}$$

where, $\widetilde{x}$ is the normalise data, x_i is the individual sample. However, if the attack falls outside the mentioned categories, we need to train the ML classifiers with the new attack patterns to adapt to emerging threats.

Table 3
The summary of generator and discriminator parameters and their corresponding values.

Generator		Discriminator	
Layer name	Value	Layer name	Value
Dense	512	Dense	1024
LeakyReLU	512	LeakyReLU	1024
BatchNormalisation	512	Dense	512
Dense	1024	LeakyReLU	512
LeakyReLU	1024	Dropout	512
BatchNormalisation	1024	Dense	256
Dense	2048	LeakyReLU	256
LeakyReLU	2048	Dropout	256
BatchNormalisation	2048	Dense	1
Dense	41		

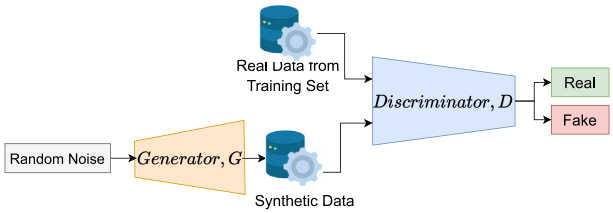


Fig. 3. Foundational structure of GANs.

3.3. Formulation of proposed GAN network

The GAN stands as an unsupervised learning model architecture within the realm of machine learning, renowned for its ability to interpret complex patterns within input data. At the core of GAN models lies their unique training methodology, known as adversarial training, as highlighted by Creswell et al. [31]. This distinctive training process propels the model's capabilities in discerning and generating complex data distributions. A schematic depiction of the typical GAN-based adversarial training procedure is presented in Fig. 3. Let $p_{\text{data}}(\mathbf{x})$ denote the true data distribution from three datasets, such as NSl-KDD, UNSW-NB15 and BoT-IoT and $p_z(\mathbf{z})$ denote the prior input noise distribution, where $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{z} \in \mathbb{R}^m$.

The objective of the GAN framework is to learn a mapping from the noise distribution $p_z(\mathbf{z})$ to the data distribution $p_{\text{data}}(\mathbf{x})$ using a generator network $G(\mathbf{z};\theta_g)$, parameterised by θ_g, where θ_g represents the weights of the generator network.

The generator network $G(\mathbf{z};\theta_g)$ aims to generate synthetic samples $\mathbf{x}$ that are indistinguishable from the true data samples.

Simultaneously, a discriminator network $D(\mathbf{x};\theta_d)$, parameterised by θ_d, is trained to distinguish between real data samples and synthetic samples generated by the generator network.

The GAN problem can be formulated as finding the optimal parameters θ_g^* and θ_d^* that minimise the following objective function:

$$\min_{\theta_g} \max_{\theta_d} V(D(\mathbf{x};\theta_d), G(\mathbf{z};\theta_g)) \tag{2}$$

where $V(D,G)$ represents the adversarial loss function defined as:

$$V(D,G) = \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})}[\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z}\sim p_z(\mathbf{z})}[\log(1 - D(G(\mathbf{z})))] \tag{3}$$

However, terminating GAN training solely based on low values of the $V(D,G)$ loss is insufficient to replicate real-world attack data accurately. Therefore, we have incorporated a boxplot distribution parameter to halt training once the boxplot distributions of synthetic ($Boxplot_S$) and real ($Boxplot_R$) data match, as determined by Eq. (5). Thus, we redefine the objective function as follows:

$$\min_{\theta_g} \max_{\theta_d} V(D(\mathbf{x};\theta_d), G(\mathbf{z};\theta_g)) \cdot \max(Boxplot_S \cong Boxplot_R) \tag{4}$$

Once Eq. (4) is satisfied, training of the GAN network will be terminated.

3.4. Synthetic data generation

In the training process, the generator and discriminator undergo iterative adjustments. The process aims to enhance the generator's capacity to produce more real-world data while simultaneously improving the discriminator's performance in differentiating between real and synthetic inputs, as shown in Fig. 3. The discriminator is trained using binary cross-entropy loss and Adam optimiser with a learning rate (α) of 0.0002. The training process is repeated for 2000 epochs (N), with a batch (B) size of 128. The proposed GAN network parameters are summarised in Table 3. The stage uses the trained generator to create data samples that

Table 4
Table of attributes excluded from UNSW-NB15, NSL-KDD, and BoT-IoT datasets during data preprocessing.

UNSW-NB15	NSL-KDD	BoT-IoT
Id	Flag	Proto
Proto	Protocol_type	Saddr
Service	Service	Daddr
State	Attack_cat	Category
Attack_cat		

Table 5
Description of various ML classifiers used in our study.

Classifier	Description
Logistic regression (LR)	A regression model that estimates the probabilities using a logistic function, widely used for binary classification tasks.
Decision tree (DT)	A tree-like model of decisions representing data splitting according to certain conditions, useful for both classification and regression.
Random forest (RF)	An ensemble learning method that operates by constructing multiple decision trees during training and outputting the mode of their predictions.
Gradient boosting (GB)	An ensemble technique that builds models sequentially, with each new model attempting to correct the errors of the previous ones.
AdaBoost (AB)	Combines multiple weak classifiers to form a strong classifier, adjusting the weights of misclassified instances.
Support vector classifier (SVC)	A representation of the data as points in space, separated into categories by a clear gap that is as wide as possible.
K-nearest neighbours (KNN)	A non-parametric method used for classification and regression, predicting the label of data point based on the majority label of its 'k' nearest neighbours.
Gaussian naive Bayes (GNB)	A probabilistic classifier based on applying Bayes' theorem with strong (naive) independence assumptions between the features.

closely mimic the characteristics of real network traffic. The synthetic data generated in this step is one of the main contributions of our study, as it plays an important role in the overall methodology. This data is meticulously labelled to match the specific attack category, with a label of 1 indicating malicious activity and a label of 0 signifying normal activity. Finally, the synthetic data is saved as a CSV file for further analysis and use in training ML models for intrusion detection. A crucial consideration in applying GAN models to NIDS is establishing the termination criteria for training. This factor significantly influences anomaly detection performance, directly impacting the quality of synthetic data used to train the detection model. Determining the termination criteria involves tracking training convergence. Monitoring training progress has typically relied on indirect methods, such as visually inspecting synthetic (generated) data. In our proposed GAN model, we have used a boxplot to see the similarity between real and synthetic data. We know that a low loss rate does not mean that it mimics the real data as expected. Therefore, we propose the synthetic data distribution for each class based on their boxplot Median, max and min values. The boxplot function can be expressed as below:

$$\text{Boxplot} = (Q1 - 1.5 \times IQR, Q1, \text{Median}, Q3, Q3 + 1.5 \times IQR) \tag{5}$$

where, Q1 is the first quartile (25th percentile), Q3 is the third quartile (75th percentile), Median is the second quartile (50th percentile), IQR is the interquartile range (Q3–Q1).

Once the real and synthetic data boxplot shows the similar pattern can terminated the training. The whole training process for generating synthetic data is shown in Algorithm 1. In this stage, we balanced the dataset before training with ML classifiers. Utilising the GAN, we generated a total of 10,000 synthetic records for each dataset involved in the study. To achieve this balance, the GAN model was employed to produce an equal number of normal and malicious traffic records, specifically 5000 of each for every dataset. Additionally, to ensure an even distribution among the various types of attacks within the malicious traffic, we generated equal quantities of records for each attack type, with the total count of all attack records summing up to 5000. This approach was consistently applied across all three datasets. Consequently, for each dataset, we compiled a total of 10,000 records, comprising an equal mix of normal and malicious traffic, which were then prepared for training with different ML classifiers. After

generating synthetic datasets, we preprocessed them to eliminate noisy data and ensure feature standardisation. Specifically, we removed certain features from the synthetic datasets (see Table 4) where most feature values were zeros across different classes.

Algorithm 1: Proposed GAN Algorithm for Synthetic Data Generation

Initialise generator network G with random weights
Initialise discriminator network D with random weights
Initialise learning rate α, number of training epochs N, and batch size B
for $epoch = 1$ to N **do**
  for $batch = 1$ to B **do**
    Sample a mini-batch of real data samples $\{x_1, x_2, ..., x_B\}$ from the dataset
    Sample a mini-batch of noise samples $\{z_1, z_2, ..., z_B\}$ from a noise distribution
    Generate fake data samples $\{G(z_1), G(z_2), ..., G(z_B)\}$ using the generator
    Train the discriminator:
      Update D using Adam:
      $\nabla_{\theta_d} \frac{1}{B} \sum_{i=1}^{B} [\log D(x_i) + \log(1 - D(G(z_i)))]$
    Sample another mini-batch of noise samples $\{z_1, z_2, ..., z_B\}$ from a noise distribution
    Generate fake data samples $\{G(z_1), G(z_2), ..., G(z_B)\}$ using the generator
    Train the generator:
      Update G using Adam:
      $\nabla_{\theta_g} \frac{1}{B} \sum_{i=1}^{B} \log(1 - D(G(z_i)))$
  end for
  Calculate Boxplot for real, $Boxplot_s$ and synthetic, $Boxplot_R$ using Eq. (5)
  score, $S = Max(Boxplot_s \cong Boxplot_R)$
  if S shows maximum value **then**
    Terminate training GAN model
  end if
end for

3.5. Machine learning classifiers for NIDS

After completing the preprocessing steps, we utilised synthetic data generated by GANs as our training dataset, while real data was reserved for testing purposes. We employed eight different classifiers to evaluate the performance of our model on the synthetic data. The classifier exhibiting the highest performance was selected as the best model. Table 5 shows a brief description of each classifier used in our study.

3.6. Evaluation metrics

Evaluation metrics provide a means to quantify the model's effectiveness in terms of its classification accuracy and error rate. The list of evaluation metrics used in this study is shown in Table 6.

In these formulas, TP denotes true positives, TN denotes true negatives, FP denotes false positives, and FN denotes false negatives. In cybersecurity, Mean Time to Detect (MTTD) refers to the average time it takes to detect a security incident or breach within an organisation's network or systems. However, our study did not use the MTTD metric because we only deployed a trained ML model for NIDS detection. The delay in testing with the trained model is low in intrusion detection.

4. Results

In this section, we discuss the outcomes of our study. Initially, we detail the evaluation metrics used to assess the performance, followed by an analysis of the performance evaluation of different ML models applied in our study.

4.1. Performance evaluation

In our study, we evaluated different classifiers across three datasets.

4.1.1. Evaluation of NIDS on UNSW-NB15 dataset

We evaluated the performance of NIDS on the UNSW-NB15 Dataset using GAN (shown in Fig. 3) as a synthetic data generator. Eight classifiers were trained for intrusion detection using synthetic data and tested with the real dataset from the UNSW-NB15 Dataset. LR achieved 75% accuracy, 75% F1 score, 75% precision, and 75% recall. DT achieved 90% accuracy, 89% F1 score, 90% precision, and 90% recall. RF achieved 68% accuracy, 55% F1 score, 78% precision, and 68% recall. GB achieved 68% accuracy, 55% F1 score, 76% precision, and 68% recall. AB achieved 68% accuracy, 55% F1 score, 53% precision, and 68% recall. SVC achieved 89% accuracy, 89% F1 score, 89% precision, and 89% recall. KNN achieved 90% accuracy, 89% F1 score, 91% precision, and 90% recall. GNB achieved 83% accuracy, 83% F1 score, 83% precision, and 83% recall. The comparison of different ML models for NIDS for UNSW-NB15 is presented in Table 7.

Table 6
Evaluation metrics for NIDs classifiers.

Evaluation metrics	Equation
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$
F1 Score	$\frac{2TP}{2TP+FP+FN}$
Precision	$\frac{TP}{TP+FP}$
Recall	$\frac{TP}{TP+FN}$

Table 7
Comparison of different ML models for NIDS for UNSW-NB15.

Classifier	Accuracy	F1 score	Precision	Recall
LR	75.00	75.00	75.00	75.00
DT	90.00	89.00	90.00	90.00
RF	68.00	55.00	78.00	68.00
GB	68.00	55.00	76.00	68.00
AB	68.00	55.00	53.00	68.00
SVC	89.00	89.00	89.00	89.00
KNN	90.00	89.00	91.00	90.00
GNB	83.00	83.00	83.00	83.00

Table 8
Comparison of different ML models for NIDS for NSL-KDD .

Classifier	Accuracy	F1 score	Precision	Recall
LR	75.00	80.00	75.00	75.00
DT	77.00	77.00	79.00	77.00
RF	75.00	75.00	75.00	75.00
GB	77.00	77.00	80.00	77.00
AB	80.00	80.00	80.00	80.00
SVC	79.00	79.00	81.00	79.00
KNN	78.00	78.00	81.00	78.00
GNB	84.00	84.00	85.00	84.00

4.1.2. Evaluation of NIDS on NSL-KDD dataset

We evaluated the performance of NIDS on the NSL-KDD Dataset using GAN (shown in Fig. 3) as a synthetic data generator. Eight classifiers were trained for intrusion detection using synthetic data and tested with the real dataset from the NSL-KDD Dataset. LR achieved 75% accuracy, 75% F1 score, 80% precision, and 75% recall. DT achieved 77% accuracy, 77% F1 score, 79% precision, and 77% recall. RF achieved 75% accuracy, 75% F1 score, 75% precision, and 75% recall. GB achieved 77% accuracy, 77% F1 score, 80% precision, and 77% recall. AB achieved 80% accuracy, 80% F1 score, 80% precision, and 80% recall. SVC achieved 79% accuracy, 79% F1 score, 81% precision, and 79% recall. KNN achieved 78% accuracy, 78% F1 score, 81% precision, and 78% recall. GNB achieved 84% accuracy, 84% F1 score, 85% precision, and 84% recall. The comparison of different ML models for NIDS for NSL-KDD is presented in Table 8.

4.1.3. Evaluation of NIDS on bot-iot dataset

We evaluated the performance of NIDs on the BoT-IoT Dataset using GAN (shown in Fig. 3) as synthetic data generators. Eight classifiers were trained for intrusion detection using synthetic data and tested with the real dataset from the BoT-IoT Dataset. LR achieved 99% accuracy, 99% F1 score, 100% precision, and 99% recall. DT achieved 99% accuracy, 100% F1 score, 100% precision, and 99% recall. RF achieved 99% accuracy, 100% F1 score, 100% precision, and 99% recall. GB achieved 99% accuracy, 100% F1 score, 100% precision, and 99% recall. AB achieved 99% accuracy, 100% F1 score, 100% precision, and 99% recall. SVC achieved 98% accuracy, 99% F1 score, 100% precision, and 98% recall. KNN achieved 100% accuracy, 100% F1 score, 100% precision, and 100% recall. GNB achieved 100% accuracy, 100% F1 score, 100% precision, and 100% recall. The comparison of different ML models for NIDS for BoT-IoT is presented in Table 9.

5. Discussion

In this section, we analyse our results and compare them with existing literature published in the field as shown in Table 10. Our study utilised synthetic data for training, with the goal of alleviating the burden on real-world datasets for NIDS. While our GAN-based synthetic data generator's training loss is low, it encounters challenges in accurately replicating real-world data. To address this limitation, we have implemented a distribution-based GAN network. Unlike conventional approaches, training in this network does not terminate until the distributions of real-world and synthetic data align, as illustrated in Fig. 4.

Fig. 4 shows that the maximum and minimum values on the boxplot are higher but indicate a closer resemblance to real data, with the median value matching. This boxplot is compelling evidence that our GAN-based synthetic data exhibits increased

Table 9
Comparison of different ML models for NIDS for BoT-IoT.

Classifier	Accuracy	F1 score	Precision	Recall
LR	99.00	99.00	100.00	99.00
DT	99.00	100.00	100.00	99.00
RF	99.00	100.00	100.00	99.00
GB	99.00	100.00	100.00	99.00
AB	99.00	100.00	100.00	99.00
SVC	98.00	99.00	100.00	98.00
KNN	100.00	100.00	100.00	100.00
GNB	100.00	100.00	100.00	100.00

Table 10
Comparison of existing methods on the datasets used in our study with our proposed method.

Reference	Dataset	Accuracy	Precision	Recall	F1 score
	UNSW-NB15	–	81.00	81.00	81.00
[20]	NSL-KDD	–	96.00	99.00	98.00
	BoT-IoT	–	99.00	99.00	99.00
[26]	NSL-KDD	91.00	87.00	98.00	92.00
[21]	NSL-KDD	–	99.00	100.00	99.00
[32]	UNSW-NB15	90.00	80.00	98.00	88.00
[33]	BoT-IoT	–	99.00	99.00	99.00
	UNSW-NB15	90.00	91.00	90.00	89.00
(Proposed)	NSL-KDD	84.00	85.00	84.00	84.00
	BoT-IoT	100.00	100.00	100.00	100.00

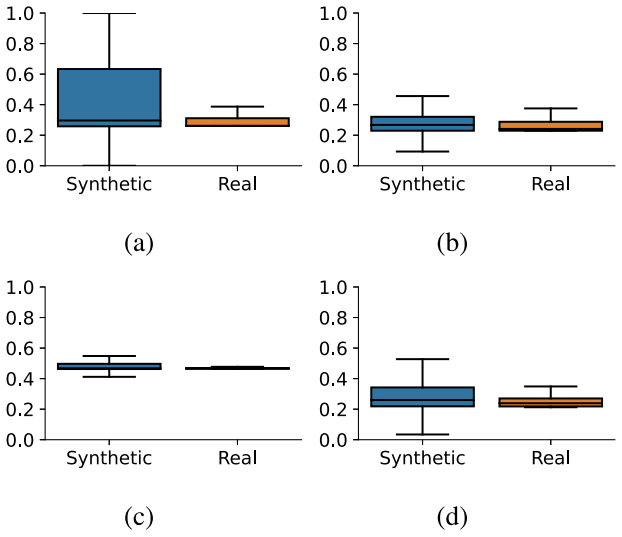


Fig. 4. Box plots were utilised to analyse features such as (a) dur, (b) dpkts, (c) sbytes, and spkts (d) in the UNSW-NB15 datasets. The presented box plots illustrate the disparities between real and synthetic data, leading to the termination of the GANs training process.

reliability in handling real-world datasets for NIDS classification tasks. By ensuring a more accurate representation of real-world data distributions, our distribution-based GAN network enhances the effectiveness of synthetic data generation for NIDS applications.

After generating synthetic datasets, we trained eight classifiers and assessed their performance across different datasets, as depicted in Figs. 5 to 7. For the UNSW-NB15 dataset, DT and KNN models achieved the highest accuracy rates (90%, 90%) compared to other classifiers. Conversely, AB and GNB exhibited superior accuracy rates (80%, 84%) compared to other models for the NSL-KDD dataset. Additionally, KNN and GNB demonstrated superior accuracy rates (100%, 100%) compared to other models for the BoT-IoT dataset. KNN shows as the top-performing model across most datasets, except for the NSL-KDD dataset. These findings validate the reliability and robustness of our proposed GAN-based synthetic data for training NIDS to detect anomalies in real-world datasets. By achieving high accuracy rates across diverse datasets, our approach emphasises its efficacy in facilitating the development of intrusion detection systems capable of accurately identifying and mitigating security threats in network environments.

Recall, as we discussed earlier, the existing studies utilising GAN have shown nearly similar results. However, they also depended on a combination of real-world and synthetic data for training. For example, Kumar et al. [20] employed XGBoost combined with

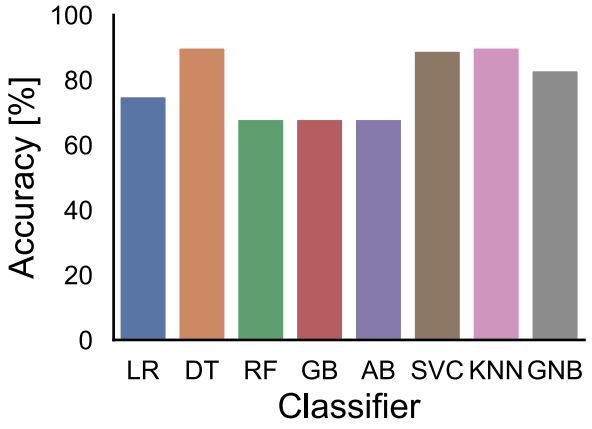


Fig. 5. Comparison of network intrusion detection systems (NIDs) classification performance using the UNSW-NB15 dataset.

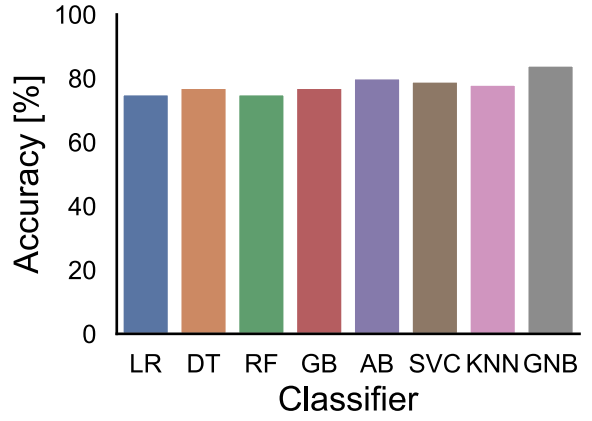


Fig. 6. Comparison of network intrusion detection systems (NIDs) classification performance using the NSL-KDD dataset.

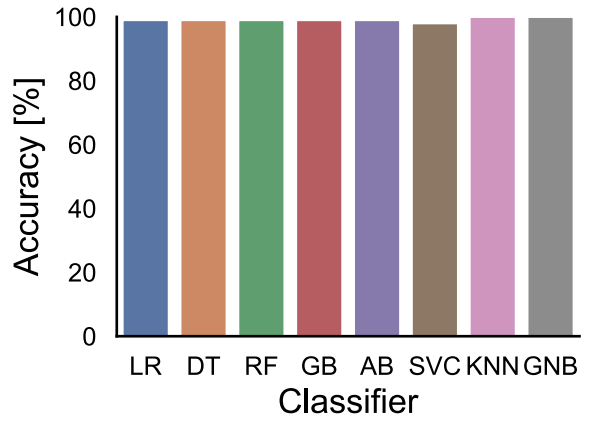


Fig. 7. Comparison of network intrusion detection systems (NIDs) classification performance using the BoT-IoT dataset.

WCGAN on the UNSW-NB15 dataset with 20% real-time training data, NSL-KDD with 15% real-time training data, and BoT-IoT with 24% real-time training data, achieving precision of 81%, 96%, 99% respectively, recall of 81%, 99%, 99% respectively, and F1 Scores of 81%, 98%, 99% respectively. Xu et al. [26] and Zhao et al. [21] both utilised the NSL-KDD dataset, achieving a precision of 87% and 99% respectively, recall of 98% and 100% respectively, and F1 scores of 92% and 99% respectively. Kasongo et al. [32] used a DT classifier for the UNSW-NB15 dataset and achieved an accuracy of 90%, precision of 80%, recall of 98%, and an F1

score of 88%. Similarly, Khanday et al. [33] employed LR as their classifier for the IoT-BoT dataset, achieving precision, recall, and F1-scores of 99%, respectively.

In contrast, our study, which exclusively used synthetic data for training purposes, demonstrated promising results as shown in Tables 7 to 9. Specifically, for the UNSW-NB15 and BoT-IoT datasets, our model's performance aligned with and exceeded existing studies. This highlights the effectiveness of our generated synthetic data. Conversely, for the NSL-KDD dataset, our results were comparable to those obtained in traditional studies. This parity in performance illustrates the robustness of our approach, which is capable of delivering competitive outcomes even in diverse datasets. Overall, our findings unfold the substantial potential of using synthetic data in network intrusion detection, offering insights into both its advantages and limitations. In addition, there is a negligible latency of the classification because the trained ML model is installed without running the GAN model to generate synthetic data.

6. Conclusion and future work

A significant constraint inherent in ML-based methodologies pertains to their dependence on real-world data for model training — a resource that is frequently limited, challenging to procure, and constrained by privacy and ethical considerations. We addressed these challenges in this paper by proposing a GAN-based framework. Our investigation showed the viability of leveraging entirely synthetic data produced through GANs to train ML models in NIDS, which is missing in the available literature. We have demonstrated the viability of using synthetic data, generated using three datasets (UNSW-NB15, NSL-KDD, and BoT-IoT), for training NIDS. Our method has shown promising results comparable to those obtained from real-time datasets, thus challenging the conventional reliance on real-world data for training NIDS.

Through a variety of experiments, our study indicated that synthetic data can effectively be used to train robust NIDS models. That said, our study achieved an accuracy of 90%, precision of 91%, recall of 90%, and an F1 score of 89% for the UNSW-NB15 dataset. For the NSL-KDD dataset, our results were 84% in accuracy, 85% in precision, 84% in recall, and 84% in F1 score. Significantly, for the BoT-IoT dataset, we attained scores of 100% across all metrics. These results are not only competitive but, in some cases, superior to those obtained using real-time data, highlighting the potential of synthetic data in this domain. In future, given the evolving nature of cyber threats, we plan to explore adaptive GAN frameworks that can dynamically adjust to emerging attack techniques. This will further ensure the continued effectiveness of our synthetic data approach in addressing the challenges associated with limited and privacy-constrained real-world data and enhancing the performance of generative models.

CRediT authorship contribution statement

Saifur Rahman: Writing – review & editing, Writing – original draft, Visualization, Methodology, Investigation, Data curation, Conceptualization. **Shantanu Pal:** Investigation, Conceptualization, Methodology, Resources, Supervision, Writing – review & editing. **Shubh Mittal:** Software. **Tisha Chawla:** Software. **Chandan Karmakar:** Conceptualization, Methodology, Project administration, Supervision, Validation, Writing – review & editing.

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

No data was used for the research described in the article.

References

[1] M. Ianculescu, D. Coardoş, O. Bica, V. Vevera, Security and privacy risks for remote healthcare monitoring systems, in: 2020 International Conference on e-Health and Bioengineering, EHB, 2020, pp. 1–4.
[2] K. Saheed, S. Henna, Heterogeneous graph transformer for advanced persistent threat classification in wireless networks, in: 2023 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), IEEE, 2023, pp. 15–20.
[3] A. Sharma, B.B. Gupta, A.K. Singh, V. Saraswat, Advanced persistent threats (APT): Evolution, anatomy, attribution and countermeasures, J. Ambient Intell. Humaniz. Comput. (2023) 1–27.
[4] T. Rincy N, R. Gupta, Design and development of an efficient network intrusion detection system using machine learning techniques, Wirel. Commun. Mob. Comput. 2021 (2021) 1–35.
[5] L. Gehri, R. Meier, D. Hulliger, V. Lenders, Towards generalizing machine learning models to detect command and control attack traffic, in: 2023 15th International Conference on Cyber Conflict: Meeting Reality, CyCon, IEEE, 2023, pp. 253–271.
[6] A. Thakkar, R. Lohiya, A review on challenges and future research directions for machine learning-based intrusion detection system, Arch. Comput. Methods Eng. (2023) 1–25.
[7] C. Yinka-Banjo, O.-A. Ugot, A review of generative adversarial networks and its application in cybersecurity, Artif. Intell. Rev. 53 (2020) 1721–1736.
[8] S. Akcay, A. Atapour-Abarghouei, T.P. Breckon, Ganomaly: Semi-supervised anomaly detection via adversarial training, in: Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part III 14, Springer, 2019, pp. 622–637.
[9] T. Vu, M. Li, H. Humayun, Y. Zhou, J. Yao, A generative adversarial network for artifact removal in photoacoustic computed tomography with a linear-array transducer, Exp. Biol. Med. 245 (7) (2020) 597–605.

[10] B. Pan, W. Zheng, et al., Emotion recognition based on EEG using generative adversarial nets and convolutional neural network, Comput. Math. Methods Med. 2021 (2021).
[11] S. Li, V. Dutta, X. He, T. Matsumaru, Deep learning based one-class detection system for fake faces generated by GAN network, Sensors 22 (20) (2022) 7767.
[12] A. Klubnikin, How much does the Internet of Things cost? 2021, ITRex, [Online] Available: https://itrexgroup.com/blog/how-much-iot-cost-factors-challenges/.
[13] N. Moustafa, J. Slay, UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set), in: 2015 Military Communications and Information Systems Conference, MilCIS, 2015, pp. 1–6.
[14] J. McHugh, Testing intrusion detection systems: A critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by lincoln laboratory, ACM Trans. Inf. Syst. Secur. 3 (4) (2000) 262–294.
[15] N. Koroniotis, N. Moustafa, E. Sitnikova, B. Turnbull, Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset, Future Gener. Comput. Syst. 100 (2019) 779–796.
[16] H. Liu, B. Lang, Machine learning and deep learning methods for intrusion detection systems: A survey, Appl. Sci. 9 (20) (2019) 4396.
[17] V. Dutta, M. Choraś, M. Pawlicki, R. Kozik, A deep learning ensemble for network anomaly and cyber-attack detection, Sensors 20 (16) (2020) 4583.
[18] J. Almaraz-Rivera, J. Perez-Diaz, A. Cantoral-Ceballos, Transport and application layer DDoS attacks detection to IoT devices by using machine learning and deep learning models, Sensors 22 (9) (2022) 3367.
[19] J. Jiang, A survey of machine learning in additive manufacturing technologies, Int. J. Comput. Integr. Manuf. (2023) 1–23.
[20] V. Kumar, D. Sinha, Synthetic attack data generation model applying generative adversarial network for intrusion detection, Comput. Secur. 125 (2023) 103054.
[21] G. Zhao, P. Liu, K. Sun, Y. Yang, T. Lan, H. Yang, Research on data imbalance in intrusion detection using CGAN, PLoS One 18 (10) (2023) e0291750.
[22] J. Lee, K. Park, GAN-based imbalanced data intrusion detection system, Pers. Ubiquitous Comput. 25 (2021) 121–128.
[23] S. Ding, L. Kou, T. Wu, A GAN-based intrusion detection model for 5G enabled future metaverse, Mob. Netw. Appl. 27 (6) (2022) 2596–2610.
[24] E. Seo, H.M. Song, H.K. Kim, GIDS: GAN based intrusion detection system for in-vehicle network, in: 2018 16th Annual Conference on Privacy, Security and Trust, PST, IEEE, 2018, pp. 1–6.
[25] H. Chen, L. Jiang, Efficient GAN-based method for cyber-intrusion detection, 2019, arXiv preprint arXiv:1904.02426.
[26] W. Xu, J. Jang-Jaccard, T. Liu, F. Sabrina, J. Kwak, Improved bidirectional gan-based approach for network intrusion detection using one-class classifier, Computers 11 (6) (2022) 85.
[27] M. Kaplan, E. Alptekin, An improved BiGAN based approach for anomaly detection, Procedia Comput. Sci. 176 (2020) 185–194.
[28] M. Chalé, N.D. Bastian, Generating realistic cyber data for training and evaluating machine learning classifiers for network intrusion detection systems, Expert Syst. Appl. 207 (2022) 117936.
[29] H. Yang, H. Yuan, L. Zhang, Risk assessment method of IoT host based on attack graph, Mob. Netw. Appl. (2023) 1–10.
[30] M. Tavallaee, E. Bagheri, W. Lu, A.A. Ghorbani, A detailed analysis of the KDD CUP 99 data set, in: 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, Ieee, 2009, pp. 1–6.
[31] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, A.A. Bharath, Generative adversarial networks: An overview, IEEE Signal Process. Mag. 35 (1) (2018) 53–65.
[32] S.M. Kasongo, Y. Sun, Performance analysis of intrusion detection systems using a feature selection method on the UNSW-NB15 dataset, J. Big Data 7 (2020) 1–20.
[33] S. Khanday, H. Fatima, N. Rakesh, Implementation of intrusion detection model for DDoS attacks in lightweight IoT networks, Expert Syst. Appl. 215 (2023) 119330.