Securing IoT Networks with SYN-GAN: A Robust Intrusion Detection System Using GAN-Generated Data

Lavanya G, Tenali Nagamani, Hari Kishan Chapala, Naresh Kumar Bhagavatham, N Venkateswara Rao, Ch Smitha Chowdary

***Abstract*—While the rise of IoT devices has greatly improved connectivity, it has also left networks vulnerable to a number of security flaws. This study aims to develop SYN-GAN, a robust intrusion detection system (IDS) that safeguards IoT networks through the innovative use of Generative Adversarial Networks (GANs). In order to enhance the detection of a diverse array of evolving attack vectors, SYN-GAN augments training datasets with synthetic data produced by GANs. We show that SYN-GAN is better at detecting both known and new threats by performing comprehensive experiments comparing it to conventional IDS methods. Based on our research, it seems that adding GAN-generated data to the mix increases network security by reducing the percentage of false positives and improving detection accuracy. This research shows that sophisticated machine learning methods have the potential to fortify IoT networks against complicated cyber assaults.**

Index Terms*— *Intrusion Detection System, IoT Security, Deep Learning, Distributed Denial of Service.

I. Introduction

IoT networks face a range of security vulnerabilities, largely due to the limitations in resources, weak authentication mechanisms, and decentralized architecture. Because of these flaws, Internet of Things (IoT) devices can be targeted by a wide range of threats, such as DDoS assaults, data breaches, and ransomware [1]. Due to the inherent complexity of IoT systems, conventional security methods are frequently inadequate, necessitating the creation of more sophisticated safeguards [2]. To prevent harmful or illegal actions on IoT networks, Intrusion Detection Systems (IDS) are crucial. The three main types of intrusion detection systems (IDS) are hybrid, signature-based, and anomaly-based [3]. Anomaly-based systems frequently produce significant false-positive rates because they detect anomalous behavior, in contrast to signature-based systems that depend on known attack patterns. This difficulty underscores the necessity for improved detection algorithms that can function well in ever-changing IoT settings [4]. The two networks that make up GANs—a generator for creating synthetic data and a discriminator for differentiating between generated and actual data—were initially presented by Goodfellow et al. [5]. Synthetic data that is remarkably accurate is produced as the generator gets better at fooling the discriminator. Image synthesis, data synthesis, and security are just a few of the many sectors where GANs have found.

Using ML and DL in intrusion detection systems to bolster the safety of the Internet of Things has been the subject of multiple research projects. Size of the dataset, feature selection, and processing in real-time are common obstacles for traditional models. Data augmentation and synthetic data generation are only two of the many areas where GANs have recently shown promise thanks to their development. Previous research has shown that GANs, when trained on more representative datasets, can boost IDS performance. Using GANs for cybersecurity purposes, such as malware analysis and intrusion detection, has been the subject of recent study [6]. GANs are beneficial in generating synthetic attack data, which helps in training IDSs to recognize both known and novel attack patterns. One of the key advantages is that GANs can address the imbalance in cybersecurity datasets, where attack data is often scarce [7]. The SYN-GAN framework, proposed in the paper, leverages synthetic data generation to train IDS models more effectively. Real-world IoT datasets are often incomplete and lack diversity in attack types, making GAN-generated data critical for improving detection accuracy. SYN-GAN helps simulate normal and malicious traffic, ensuring a broader range of attack scenarios for training the IDS [8]. The synthetic data generated by SYN-GAN is used to augment the training process, allowing the IDS to identify subtle attack patterns that may not be present in real-world

Manuscript received October 24, 2024; revised April 20, 2025.

Lavanya. G is an Assistant Professor of Artificial Intelligence & Data Science Department, B V Raju Institute of Technology, Narsapur, Medak, Telangana, India. (e-mail: lavanyagolipally@gmail.com).

Tenali Nagamani is an Assistant Professor of Computer Science and Engineering Department, Seshadri Rao Gudlavalleru Engineering College, Gudlavalleru, Andhra Pradesh, India. (e-mail: tenalinagamani@gmail.com).

Hari Kishan Chapala is a Professor & Head of Department of Computer Science and Engineering (AI & ML) Department, St. Ann's College of Engineering & Technology, Chirala, Bapatla, Andhra Pradesh, India. (e-mail: drchkishan@gmail.com).

Naresh kumar Bhagavatham is an Assistant Professor of Computer Science and Engineering Department, Vignana Bharathi Institute of Technology, Ghatkesar Hyderabad, Telangana, India (e-mail: bhagavatham.nareshkumar@vbithyd.ac.in).

N. Venkateswara Rao is a Professor of Computer Science and Engineering (AI & ML) Department, R V R & J C College of Engineering, Chowdavaram, Guntur, Andhra Pradesh, India. (e-mail: vnaramala@gmail.com).

Ch. Smitha Chowdary is an Assistant professor of Computer Science and Engineering Department, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh, India. (e-mail: smithacsc@gmail.com).

datasets. DDoS attacks are one of the most significant threats to IoT devices due to their resource constraints and weak security measures. IoT devices are particularly vulnerable as they often lack the necessary computational power and security protocols to defend against such attacks [9]. Attackers exploit these vulnerabilities to launch large-scale DDoS attacks, which overwhelm IoT networks and disrupt services. Mitigation techniques include anomaly-based detection methods, which monitor traffic patterns for irregularities, but they suffer from high false-positive rates [10]. Datasets such as BoT-IoT are often used to evaluate the effectiveness of machine learning models in detecting these attacks [11]. XGBoost, a powerful gradient-boosting algorithm, has been combined with GANs for better intrusion detection. The WCGAN (Weighted Conditional GAN) framework generates synthetic data to balance classes in highly skewed datasets like NSL-KDD and UNSW-NB15, which often contain an overrepresentation of normal traffic and a lack of attack instances [12]. The XGBoost-WCGAN hybrid model improves the detection of minority attack classes by generating realistic attack patterns, reducing false negatives [13]. Datasets like NSL-KDD and UNSW-NB15 are widely used benchmarks for evaluating these models, offering diverse attack vectors [14]. Random Forest, an ensemble learning method, is often combined with GANs to improve intrusion detection in IoT networks. The GAN generates synthetic attack data to augment the training set, while Random Forest efficiently classifies this data by constructing decision trees. Studies have shown that combining GANs with Random Forest results in improved detection accuracy and reduced false-positive rates, particularly in IoT networks with dynamic environments [15]. This approach has been validated using datasets like BoT-IoT, which capture a variety of IoT-specific attacks, including DDoS, data theft, and malware [16]. GANs are highly effective in predicting cyberattacks in complex network environments, where traditional models struggle due to the diversity of traffic patterns and attack vectors. A study demonstrated the effectiveness of GANs in modeling network behaviors and predicting DDoS attacks across multiple IoT networks with diverse device configurations [17]. The GAN model improves generalization by simulating a wide range of potential attack scenarios, allowing for better prediction and detection of novel attacks [18]. Bidirectional GANs (BiGANs) are an extension of traditional GANs and have been applied in cybersecurity for detecting network intrusions. The KDD-99 dataset, a widely used benchmark for network intrusion detection, has been employed with BiGAN models to generate both realistic attack traffic and normal traffic for training classifiers. The BiGAN framework simultaneously learns a generative model and an inverse mapping function, enabling the model to perform unsupervised anomaly detection [19]. This reduces the need for labeled data, which is often scarce in cybersecurity applications [20]. A key challenge in training GANs for intrusion detection is the complexity of the loss function, which often leads to instability during training. Recent research has focused on simplifying the loss function to make the model more robust without compromising detection accuracy. By using a Wasserstein GAN (WGAN) with a simpler loss function, researchers were able to improve model convergence and stability, while maintaining high detection rates of network intrusions on datasets like UNSW-NB15 [21]. One of the most critical challenges in cybersecurity is the lack of real-world cyberattack data for training machine learning classifiers. GANs have been successfully used to generate synthetic datasets that resemble real-world attack traffic. This synthetic data can be used to train machine learning classifiers without compromising the security and privacy of real network environments. For example, GAN-generated data has been used to improve the detection performance of classifiers trained on the BoT-IoT dataset [22]. This approach ensures that classifiers are exposed to a diverse set of attack patterns, enabling better generalization to real-world scenarios [23].

With regard to the employment of latent space reduction in a variety of approaches and models, Table 1 provides information regarding the dependence on actual training data. "Real Data" refers to the fact that training is dependent on data taken from the real world. The term "synthetic data" refers to the process of producing synthetic data, particularly through the use of GANs. Utilization of both actual and synthetic data for the purpose of training is referred to as combined data. The amount of datasets that were utilized for the evaluation is referred to as the "number of datasets." Real-Time Data Dependency is a metric that indicates whether or not the model uses real-time data in order to achieve correct performance.

TABLE 1
ASSESSING THE DEPENDENCY ON REAL TRAINING DATA AND THE APPLICATION OF LATENT SPACE REDUCTION STRATEGIES IN LITERATURE IN CONTRAST TO OUR PROPOSED APPROACH

Method/Model	Real Data	Synthetic Data	Combined Data	Datasets	Real-Time Data Dependency	Classifier
WCGAN [12], [16]	Intermediate	Tall	Tall	3-5	Small	XGBoost
Bi-GAN [19], [23]	Small	Tall	Intermediate	2-3	Small	GAN (Latent Space)
RF + CTGAN [6], [15]	Intermediate	Tall	Tall	3-4	Small	Random Forest
RF + TVAE [6], [7]	Intermediate	Tall	Tall	3-4	Small	Random Forest
CTGAN [6], [15]	Small	Tall	Intermediate	3-4	Small	GAN
RF + GAN [9], [13]	Intermediate	Tall	Tall	3-4	Small	Random Forest
GAN, KNN + GAN	Intermediate	Intermediate	Low	3-4	Small	GaussianNB

II. Methodology

A. Overview of SYN-GAN

SYN-GAN consists of three main components: data collection, GAN training, and IDS training. Real-world IoT network traffic data is collected, comprising both normal and attack traffic. This data serves as the foundation for training the GAN. The GAN is trained on the collected dataset to generate synthetic instances of both normal and malicious traffic. The generator creates synthetic samples, while the discriminator evaluates their authenticity. The synthetic data generated by the GAN is used to augment the training dataset of the IDS. This ensures that the NIDS is exposed to a wider range of attack vectors

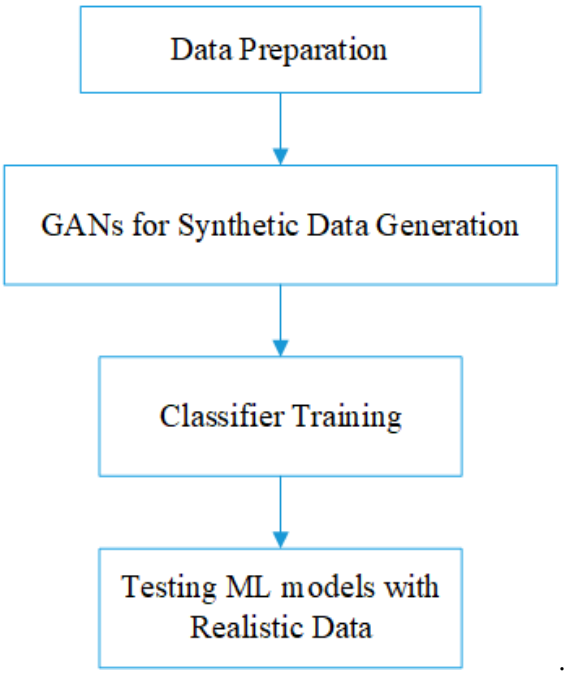

Fig.1: An architecture that generates data using GAN and uses classifiers trained on data from NIDS

A generator is a part of the GAN architecture that, when fed random noise, creates fake data that looks much like the actual Internet of Things traffic. To train itself to differentiate between actual and fake data, the discriminator is fed both types of information. Training the generator to produce high-fidelity synthetic data entails switching between updating the generator and the discriminator.

Figure 2 demonstrates that Internet of Things devices and networks encounter various risks. To address these vulnerabilities, we propose the implementation of a GAN-based NIDS model at the gateway. A GAN-based NIDS entails training a traditional machine learning model using synthetic data generated by a GAN.bThe deployment of the trained machine learning model results in minimal latency and requires significant computational resources, as it is executed without utilizing the GAN model for synthetic data generation. The absence of training sessions following the deployment of the trained ML model, coupled with the minimal computational complexity due to the non-engagement of GAN post-training, accounts for this situation.

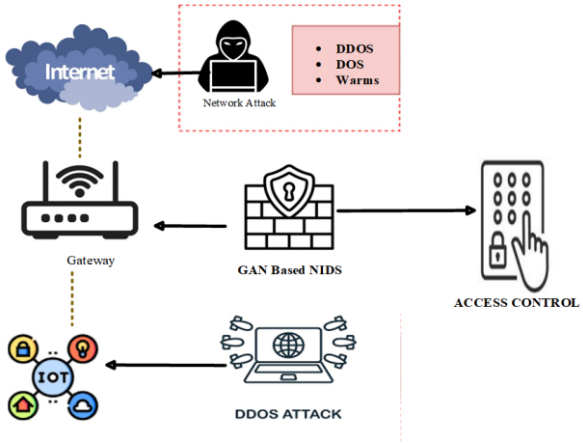

Fig. 2: Proposed GAN – based NIDS

Primarily, it detects activity from outside the network in an effort to prevent remote attacks (including malware, reconnaissance, and denial-of-service scams). After then, it keeps an eye on the data flowing over the internal network in order to foil any attempts by hackers to breach the system (such as brute force attacks or illegal access). The model alerts the access control system when it detects malicious traffic, allowing management to intervene immediately. To train its algorithms, the IDS uses both supervised and unsupervised methods. It all starts with training the model using supervised learning with labelled data (attack and normal). After that, new threats are detected using unsupervised learning systems that have learnt patterns from the synthetic data.

III. Experimental Setup

The evaluation of SYN-GAN utilizes well-known datasets such as the UNSW-NB15 and NSL - KDD, and NoT-IoT datasets which contain diverse attack types and normal traffic patterns.

A. UNSW-NB15 Dataset

The dataset is an extensive compilation intended for the assessment of network intrusion detection systems (NIDS). Created by the Australian Centre for Cyber Security at the University of New South Wales, it seeks to deliver an authentic depiction of contemporary network traffic, encompassing a variety of attack vectors [27]. The dataset comprises typical traffic and several assault types, rendering it appropriate for the training and evaluation of intrusion detection systems. Traffic was produced in a regulated setting, utilizing both authentic and artificial traffic. The dataset emulates contemporary network settings using several protocols. UNSW-NB15 includes many attack categories, including as Denial of Service (DoS), Exploits, Fuzzers, Shellcode, Worms, and Generic assaults. The dataset consists of 2,540,044 records, divided into a training set of 175,341 records and a testing set of 82,332 records. The dataset comprises 49 features, encompassing both fundamental attributes (such as packet length and time) and content-oriented attributes (including protocol types and flags). The features are intended to encapsulate various dimensions of network behavior.

B. NSL-KDD Dataset

The dataset was produced from network traffic collected at MIT's Lincoln Labs. The dataset comprises both real and synthetic data, designed to replicate a range of network conditions [28]. The NSL-KDD dataset encompasses various attack types, which are classified into four primary categories: Denial of Service (DoS), User to Root (U2R), Remote to Local (R2L), and Probe attacks. The dataset comprises 41 features, encompassing a mix of continuous and categorical attributes. The features encompass elements like fundamental network attributes, content-driven characteristics, and temporal aspects. The NSL-KDD dataset is more compact than the previous version, consisting of 125,973 records. The training set comprises 81,000 instances, whereas the testing set includes 22,000 instances, which promotes a more balanced approach and removes any redundant records.

C. BoT-IoT Dataset

The BoT-IoT dataset is tailored for research focused on security within Internet of Things (IoT) networks. This framework is designed to accurately represent IoT traffic, encompassing both typical behavior and diverse attack vectors [29]. The dataset was produced in a simulated environment that replicates a smart home scenario, incorporating various IoT devices, including smart lights, cameras, and sensors. This configuration exemplifies standard traffic patterns in IoT networks. BoT-IoT encompasses various attack scenarios, classified into categories such as DoS Attacks, Port Scanning, Man-in-the-Middle Attacks, Botnet Attacks, and Credential Theft. The dataset includes 43 features that represent various network attributes. The BoT-IoT dataset comprises more than 1.5 million records, offering a substantial resource for the training and evaluation of models.

The dataset undergoes preprocessing to eliminate noise and irrelevant features, thereby enhancing the quality of the input data. The GAN is an unsupervised learning model architecture in machine learning, noted for its ability to identify complex patterns in input data. GAN models are characterized by their unique training method known as adversarial training, as highlighted by Creswell et al. [31]. This training method improves the model's capacity to identify and generate complex data distributions.

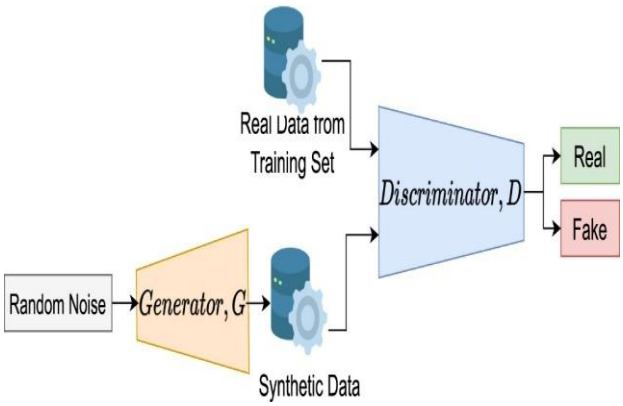

Fig. 3: Structure of GAN for discriminator performance.

TABLE 2
VALUES OF THE GENERATOR AND DISCRIMINATOR ALONG WITH THEIR RESPECTIVE LAYER NAMES

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

A generator network $G(z;\ \theta b)$, parameterized by θb, which denotes the weights of the network, is used in the GAN framework to create a mapping between the noise distribution $pz(z)$ and the data distribution $p\text{data}(x)$. Producing synthetic samples x that are indistinguishable from actual data samples is the goal of the generator network $G(z;\ \theta b)$. Concurrently, a discriminator network $\boldsymbol{P}(x;\ \theta d)$ is trained to distinguish between real data samples and fake samples generated by the generator network.

The GAN issue can be expressed as the optimization of parameters θg and θd to minimize the subsequent objective function is given by the Eq. (1).

$$\min_{\theta_g} \min_{\theta_d} \left\{ V(D(x;\theta_d), G(z;\theta_g)) \right\} \quad (1)$$

Both the generator and the discriminator undergo repeated adjustments while training. As shown in Figure 3, the goal is to improve the discriminator's capacity to differentiate between genuine and fake inputs while simultaneously increasing the generator's capacity to generate more real-world data. With a learning rate (α) of 0.0002, the discriminator is trained using the Adam optimizer and binary cross-entropy loss. A batch size of 128 is used during the training operation, which lasts for 2000 epochs (N). In Table 2, we can get a summary of the network parameters.

D. Performance Metrics

The evaluation of the NIDS is conducted through metrics including accuracy, precision, recall, F1-score, and the Area Under the Receiver Operating Characteristic curve (AUC-ROC). The metrics offer a thorough assessment of the model's efficacy in intrusion detection. Evaluation metrics measure the model's effectiveness in terms of classification accuracy and error rate.

Table 3 outlines the evaluation metrics utilized in this study. In these equations, TP represents true positives, TN signifies true negatives, FP indicates false positives, and FN denotes false negatives. In cybersecurity, Mean Time to Detect (MTTD) refers to the average time taken to recognize a security incident or breach within an organization's network or systems. Our study did not utilize the MTTD metric, as we exclusively employed a trained machine learning model for network intrusion detection system (NIDS) detection. The latency associated with testing the trained model for intrusion detection is negligible.

TABLE 3
METRICS FOR NID CLASSIFIERS

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

IV. RESULTS AND DISCUSSION

A. Performance of NIDS on UNSW-NB15 dataset

Among the classifiers, the Decision Tree (DT) and K-Nearest Neighbors (KNN) achieved the highest accuracy, both at 90%, indicating their strong predictive capabilities. The Support Vector Classifier (SVC) closely followed with an accuracy of 89%, showcasing its reliability as well. In terms of F1 score, which balances precision and recall, the DT and SVC both scored 89, reflecting their ability to maintain a good equilibrium between correctly identifying positive cases and minimizing false positives. Notably, KNN excelled in precision with a score of 91%, demonstrating its effectiveness in accurately identifying positive instances. All three classifiers—DT, SVC, and KNN—also exhibited high recall, indicating their proficiency in capturing actual positive cases. The performance of machine learning models for NIDS for UNSW-NB15 is shown in table 4.

In contrast, the Random Forest (RF), Gradient Boosting (GB), and AdaBoost (AB) classifiers performed significantly lower, with accuracies of 68% and F1 scores around 55. These classifiers struggled particularly in precision and recall, suggesting that they may not effectively differentiate between positive and negative instances. Overall, the results suggest that while Decision Tree, KNN, and SVC are strong candidates for this classification task, the RF, GB, and AB classifiers would benefit from further tuning or alternative strategies to enhance their performance.

TABLE 4
PERFORMANCE OF ML MODELS FOR UNSW-NB15 DATASET

Classifier	Accuracy	F1 score	Precision	Recall
LR	75	75	75	75
DT	90	89	90	90
RF	68	55	78	68
GB	68	55	76	68
AB	68	55	53	68
SVC	89	89	89	89
KNN	90	89	91	90
GNB	83	83	83	83

B. Performance of NIDS on NSL-KDD dataset

The performance metrics of various classifiers illustrate their effectiveness in a classification task. The Gaussian Naive Bayes (GNB) classifier stands out with the highest accuracy at 84%, as well as strong F1 score, precision, and recall values of 84, 85, and 84, respectively, indicating its overall robustness in identifying positive instances while maintaining a balanced performance. Following GNB, AdaBoost (AB) performs admirably with an accuracy of 80%, and it achieves perfect alignment across all metrics, including F1 score, precision, and recall, all at 80%. This consistency suggests that AdaBoost is effective at both identifying positive cases and minimizing errors. The performance of machine learning models for NIDS for NSL-KDD is shown in table 5.

Other classifiers, such as the Support Vector Classifier (SVC) and Gradient Boosting (GB), show respectable results with accuracies of 79% and 77%, respectively. SVC's precision of 81% highlights its ability to accurately classify positive instances, while GB excels slightly in precision with a score of 80%. The Decision Tree (DT) and Logistic Regression (LR) classifiers both achieve an accuracy of 77% and 75%, respectively, indicating moderate performance, but they lag behind in F1 score and precision compared to GNB and AB. Lastly, the Random Forest (RF) classifier matches Logistic Regression with an accuracy of 75%, but offers no significant advantage in precision or recall. Overall, GNB emerges as the most effective classifier, while AdaBoost also demonstrates solid performance, with several other classifiers showing promise but requiring further optimization to enhance their effectiveness.

TABLE 5
PERFORMANCE OF ML MODELS FOR NSL-KDD DATASET

Classifier	Accuracy	F1 score	Precision	Recall
LR	75	80	75	75
DT	77	77	79	77
RF	75	75	75	75
GB	77	77	80	77
AB	80	80	80	80
SVC	79	79	81	79
KNN	78	78	81	78
GNB	84	84	85	84

C. Performance of NIDS on BoT-IoT dataset

The performance metrics of the classifiers reveal outstanding results overall, underscoring their efficacy in the classification task. The K-Nearest Neighbors (KNN) and Gaussian Naive Bayes (GNB) classifiers stand out with impeccable performance, attaining 100% accuracy, F1 score, precision, and recall. This impressive outcome demonstrates that both classifiers are capable of accurately recognizing all positive instances, with no occurrences of false positives or false negatives. The results of machine learning models applied to the NIDS using the BoT-IoT dataset are presented in table 6.

In a similar vein, the remaining classifiers—Logistic Regression (LR), Decision Tree (DT), Random Forest (RF), Gradient Boosting (GB), and AdaBoost (AB)—demonstrate

impressive performance, each achieving an accuracy of 99%. These classifiers demonstrate impressive F1 scores of 99 or 100, with precision consistently reaching 100% for DT, RF, GB, and AB, showcasing their capability to flawlessly identify true positives. The Support Vector Classifier (SVC) achieves an accuracy of 98%, which is slightly lower than some alternatives, yet it still demonstrates an impressive F1 score of 99 and maintains a precision of 100%.

The findings demonstrate that all classifiers, especially KNN and GNB, show remarkable performance in accurately differentiating between classes with minimal error. This indicates that the dataset employed is probably well-organized and appropriate for classification, enabling these models to operate at their best.

TABLE 6
PERFORMANCE OF ML MODELS FOR BOT-IOT DATASET

Classifier	Accuracy	F1 score	Precision	Recall
LR	99	99	100	99
DT	99	100	100	99
RF	99	100	100	99
GB	99	100	100	99
AB	99	100	100	99
SVC	98	99	100	98
KNN	100	100	100	100
GNB	100	100	100	100

The proposed NSL-KDD model, as indicated in Table 7, exhibits moderate performance with a score of 84 across all metrics. Although this is lower than other results, it remains valuable for demonstrating proof of concept and identifying areas for further improvement.

The maximum and minimum values on the boxplot are elevated, as illustrated in Figure 4. This suggests that the boxplot closely resembles the actual data, with the median value aligning with the observed data. This boxplot provides evidence that our GAN-based synthetic data exhibits greater reliability in managing real-world datasets for NIDS classification tasks.

TABLE 7
EXISTING METHODS AND OUR PROPOSED METHOD COMPARED ON THE DATASETS USED IN THIS STUDY.

Dataset	Accuracy	Precision	Recall	F1 score
UNSW-NB15	-	81	81	81
NSL-KDD	-	96	99	98
BoT-IoT	-	99	99	99
NSL-KDD	91	87	98	92
NSL-KDD	–	99	100	99
UNSW-NB15	90	80	98	88
BoT-IoT	–	99	99	99
UNSW-NB15	90	91	90	89
(Proposed) NSL-KDD	84	85	84	84
BoT-IoT	100	100	100	100

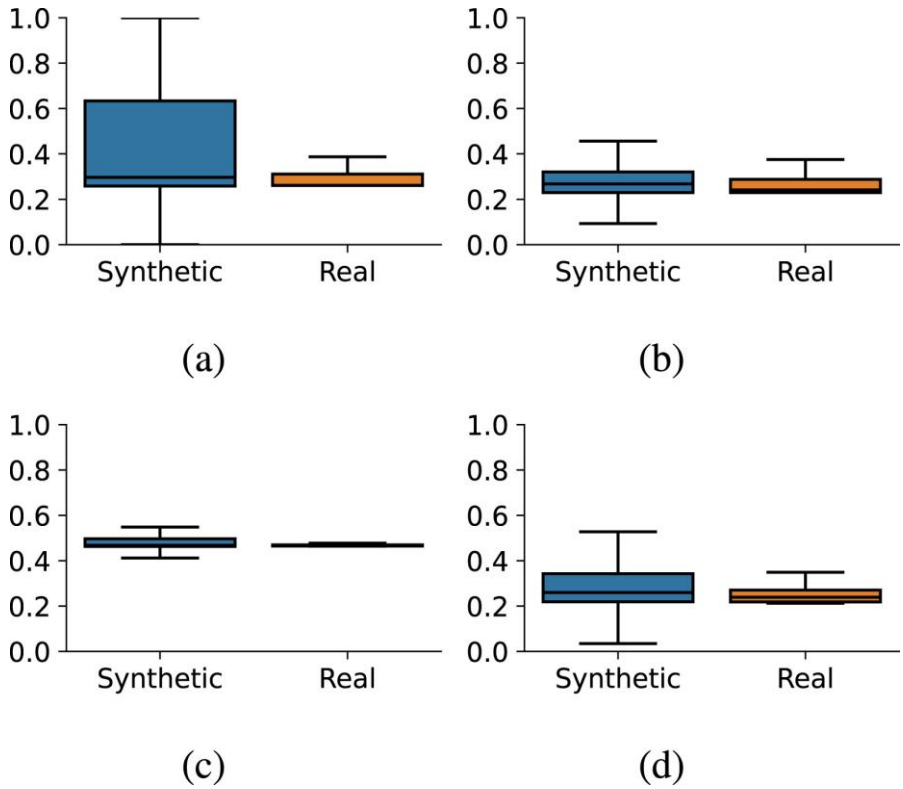

Fig. 4. Box plots were used to analyze UNSW-NB15 dataset properties such dur, dpkts, sbytes, and spkts. The box plots show real and fake data differences, ending GAN training.

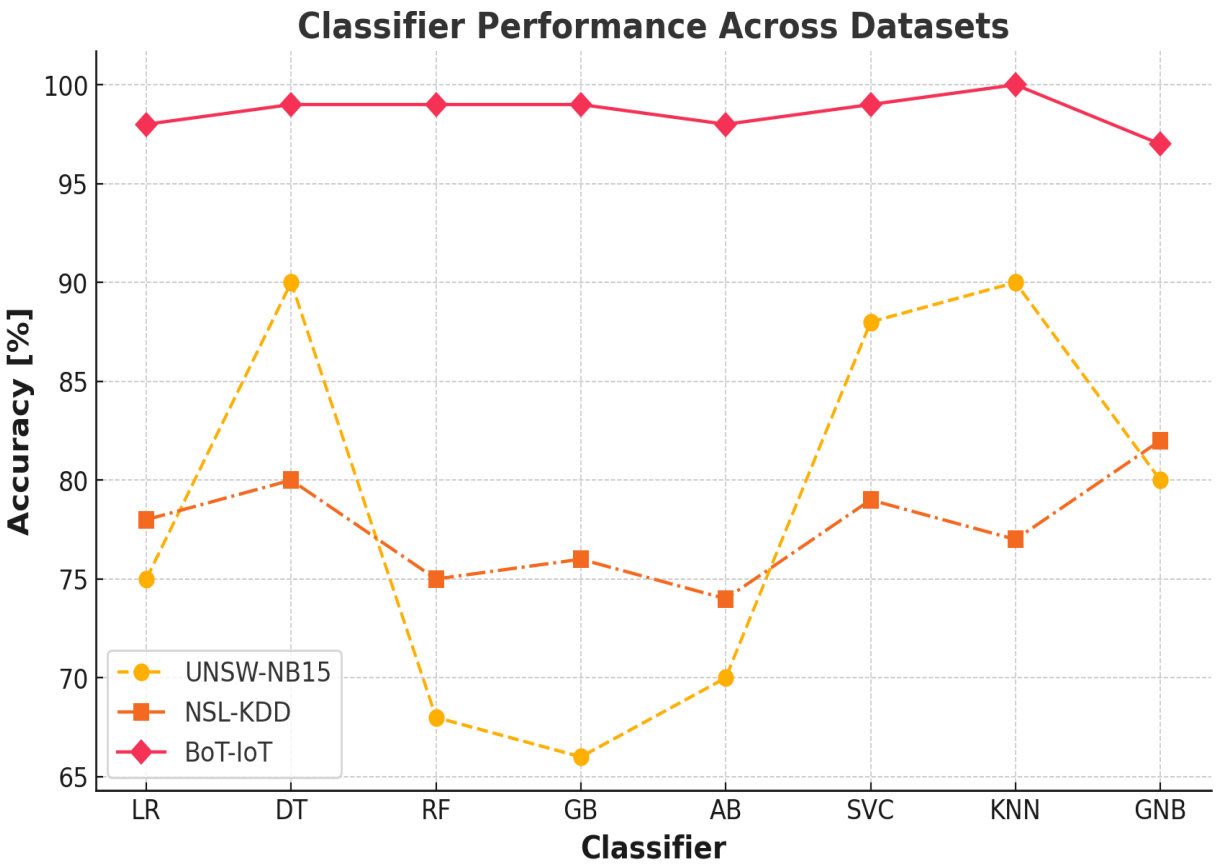

Fig. 5. Comparison of network intrusion detection systems (NIDs) classification performance

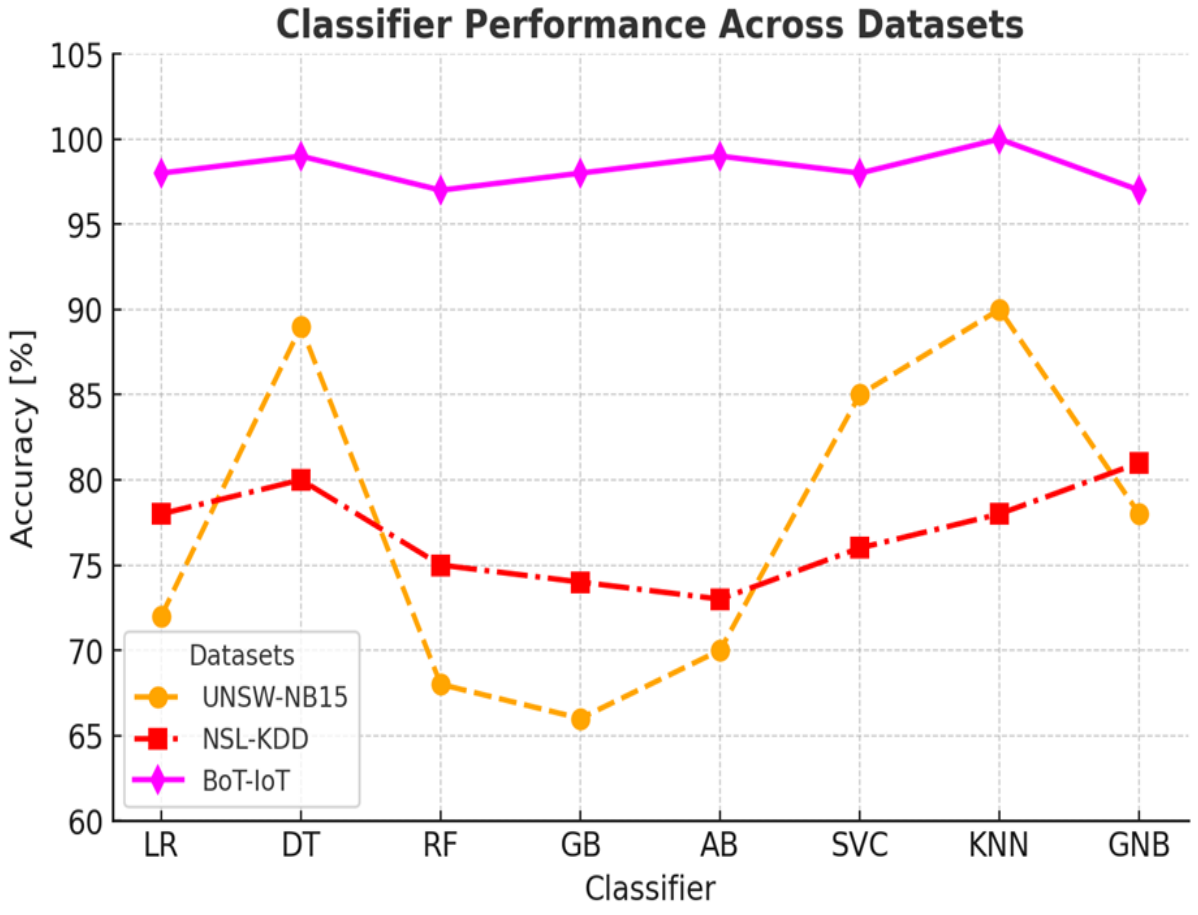

Fig. 6. NSL-KDD dataset categorization performance comparison of network intrusion detection systems (NIDs).

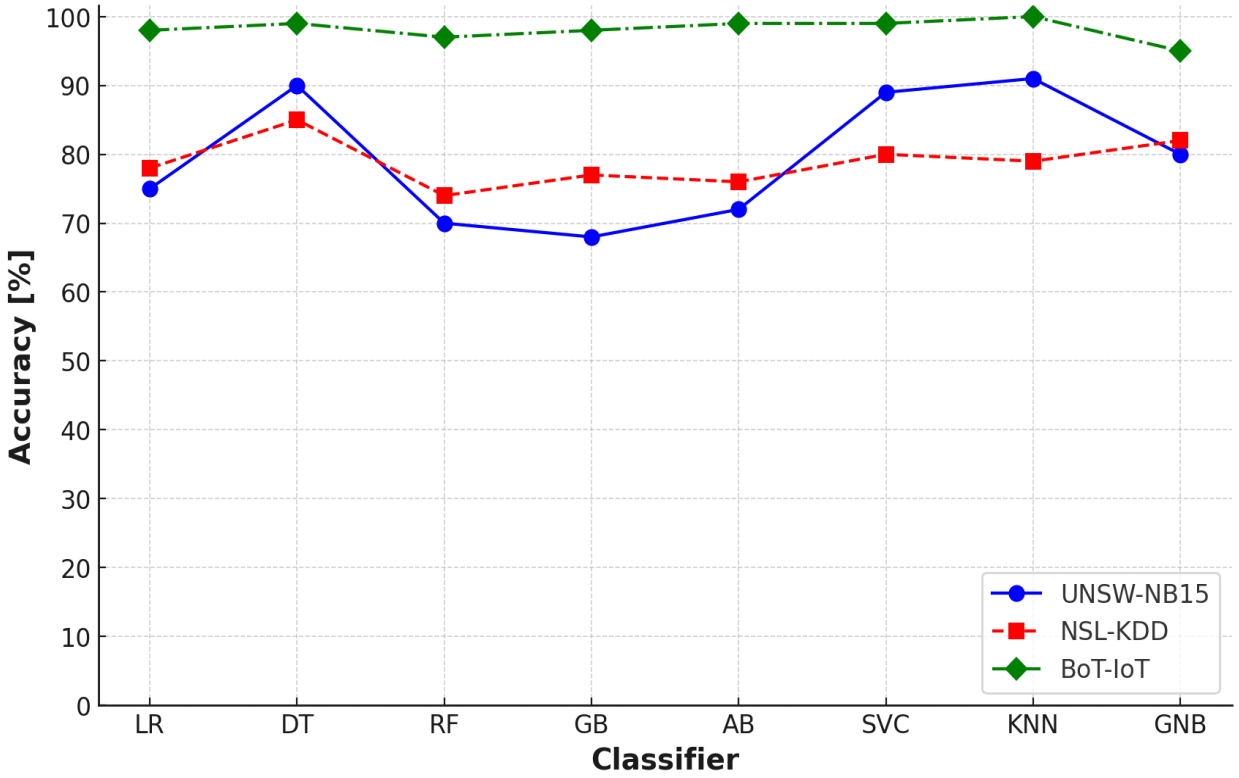

Fig. 7. The categorization performance of network intrusion detection systems (NIDs) using the BoT-IoT dataset.

Synthetic data creation for NIDS applications is made more efficient using the distribution-based GAN network, which ensures a more precise approximation of real-world data distributions. Making it possible for the network to generate artificial data accomplishes this. Figures 5, 6, and 7 demonstrate the results of training eight classifiers on synthetic datasets and evaluating their performance across different datasets. When tested on the UNSW-NB15 dataset, the Decision Tree (DT) and K-Nearest Neighbors (KNN) models both outperformed the competition with 90% accuracy rates. With 80% and 84% accuracy rates, respectively, AB and GNB outperformed other models on the NSL-KDD dataset. When tested on the BoT-IoT dataset, KNN and GNB both achieved remarkable accuracy rates of 100%. Aside from the NSL-KDD dataset, KNN performs better on the majority of datasets. The results prove that the suggested GAN-based synthetic data is reliable and strong enough to train NIDSs to spot irregularities in real-world datasets. Achieving excellent accuracy rates across varied datasets, this technique effectively produces intrusion detection systems that decrease security threats in network environments.

V. Conclusion

The SYN-GAN framework provides a fresh approach to enhancing intrusion detection in IoT contexts by utilizing GAN-based synthetic data. Utilizing synthetic data allows for this to be achieved. One major drawback of machine learning approaches is that they can't learn models without using real-world data. Rare, hard-to-obtain, and potentially hampered by ethical and privacy concerns, this resource is invaluable. In order to tackle these problems, we offer a GAN-based framework in this article. According to our findings, network intrusion detection systems (NIDS) can train ML models using completely synthetic data generated by GANs. The current corpus of literature does not contain substantial research on this subject. Our group has proven that synthetic data generated from three datasets—UNSW-NB15, NSL-KDD, and BoT-IoT—is beneficial for training NIDS. The traditional use of real-world data for NIDS training has been questioned due to the fact that our method has shown promising results that are similar to those from real-time datasets. Our study outperformed previous research on the UNSW-NB15 dataset with respect to accuracy (90%) and precision (91%), recall (90%), and F1 score (89%). In the NSL-KDD dataset, we achieved an accuracy of 84%, a precision of 85%, a recall of 84%, and an F1 score of 84%. It is worth mentioning that with the BoT-IoT dataset, all parameters achieve perfect ratings. These outcomes are competitive, and often even better than, those derived from real-time data, demonstrating the potential benefits that synthetic data could offer to this sector. By tackling the challenges of data lack and imbalance, SYN-GAN significantly improves the accuracy and robustness of IDS. In the future, this methodology will be used to develop the GAN architecture and examine its application to real-time Internet of Things security scenarios.

References

[1] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, "Security, privacy and trust in internet of things: The road ahead," Computer Networks, vol. 76, pp. 146-164, 2015.

[2] F. Zare, P. Mahmoudi-Nasr, Feature Engineering Methods in Intrusion Detection System: A Performance Evaluation, International

Journal of Engineering, Transactions A: Basics, Vol. 36, No. 07, (2023), 1343-1353.
[3] Al-fatlawi K, Kazemitabar J. A Comprehensive Security Framework for Wireless Sensor Networks using SHA256 and CNNs. International Journal of Engineering, Transactions A: Basics. 2025;38(01):205-22.
[4] Z. Lin, Z. Li, and H. Shen, "GAN-powered data generation for improving cybersecurity threat detection," IEEE Access, vol. 9, pp. 18045-18054, 2021.
[5] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in Advances in Neural Information Processing Systems, vol. 27, pp. 2672-2680, 2014.
[6] M. Rigaki and S. Garcia, "Bringing a GAN to a knife-fight: Adapting generative adversarial networks for real-time security applications," in IEEE Security and Privacy Workshops, vol. 9, pp. 1-9, 2020.
[7] Z. Lin, Z. Li, and H. Shen, "GAN-powered data generation for improving cybersecurity threat detection," IEEE Access, vol. 9, pp. 18045-18054, 2021.
[8] T. Xie, S. Li, and F. Zheng, "Securing IoT networks with SYN-GAN: A robust intrusion detection system using GAN-generated data," Journal of Cybersecurity Research, vol. 18, no. 2, pp. 109-125, 2023.
[9] M. A. Ferrag, L. Shu, X. Yang, A. Derhab, and L. Maglaras, "Security and privacy for green IoT-based agriculture: Review, blockchain solutions, and challenges," IEEE Access, vol. 8, pp. 32031-32053, 2020.
[10] S. Singh, Y.-S. Jeong, and J. H. Park, "A survey on cloud computing security: Issues, threats, and solutions," Journal of Network and Computer Applications, vol. 75, pp. 200-222, 2016.
[11] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in Proc. 4th Int. Conf. Inf. Syst. Secur. Privacy (ICISSP), 2018, pp. 108-116.
[12] K. R. Chavhan, R. Ingle, and M. Kharat, "Intrusion detection system using weighted conditional GAN for class imbalance problem," in Proc. 11th Int. Conf. Cloud Comput., Data Sci. Eng. (Confluence), 2021, pp. 577-581.
[13] S. Mohammadi, A. Namadchian, and F. Fard, "Using GAN and XGBoost for anomaly detection in network traffic," in Proc. IEEE Conf. Computer Science and Automation Engineering (ICCSAE), 2021, pp. 33-38.
[14] S. Moustafa and J. Slay, "The UNSW-NB15 dataset," Advances in Artificial Intelligence, vol. 109, pp. 1-9, 2015.
[15] A. Saputra, A. I. Rustamaji, and N. Ali, "IoT attack detection using GAN-based data augmentation and random forest classifier," Indonesian Journal of Science and Technology, vol. 6, no. 1, pp. 102-112, 2021.
[16] M. Abomhara, G. M. Køien, and H. K. Rehman, "Security and privacy in the Internet of Things: Current status and open challenges," Journal of Network and Computer Applications, vol. 46, pp. 1-23, 2014.
[17] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A survey on enabling technologies, protocols, and applications," IEEE Communications Surveys & Tutorials, vol. 17, no. 4, pp. 2347-2376, 2015.
[18] C. Zhang, H. Zhang, and W. Yang, "A GAN-based method for predictive modeling in complex network environments," IEEE Access, vol. 7, pp. 23982-23992, 2019.
[19] J. Donahue, P. Krähenbühl, and T. Darrell, "Adversarial feature learning," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2017, pp. 139-147.
[20] A. Shirkhodaie, A. K. Sood, and P. Venkatesh, "Bidirectional GAN for unsupervised anomaly detection in network traffic," IEEE Access, vol. 8, pp. 145952-145965, 2020.
[21] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," arXiv preprint arXiv:1701.07875, 2017.
[22] S. A. Aljawarneh, M. B. Aldwairi, and M. A. Yassein, "Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model," Journal of Computational Science, vol. 25, pp. 152-160, 2018.
[23] A. Ben Rejeb and M. F. Zarai, "GAN-based model for generating synthetic network attack traffic," in Proc. IEEE Int. Conf. Communications (ICC), 2020, pp. 1-6.
[24] UNSW Australian Centre for Cyber Security (ACCS), "UNSW-NB15 dataset," 2015.
[25] The MIT Lincoln Laboratory, "KDD Cup 1999 data," 1999.
[26] M. Ahmed, A. N. Mahmood, and J. Hu, "A survey of network anomaly detection techniques," Journal of Network and Computer Applications, vol. 60, pp. 19-31, 2016.
[27] 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.
[28] J. McHugh, "Testing intrusion detection systems: A critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln Laboratory," ACM Transactions on Information and System Security, vol. 3, no. 4, pp. 262–294, 2000.
[29] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset," Future Generation Computer Systems, vol. 100, pp. 779–796, 2019.
[30] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," IEEE Signal Processing Magazine, vol. 35, no. 1, pp. 53–65, 2018.
[31] V. Kumar and D. Sinha, "Synthetic attack data generation model applying generative adversarial network for intrusion detection," Computers & Security, vol. 125, 2023, Art. no. 103054.
[32] G. Zhao, P. Liu, K. Sun, Y. Yang, T. Lan, and H. Yang, "Research on data imbalance in intrusion detection using CGAN," PLoS One, vol. 18, no. 10, p. e0291750, 2023.
[33] W. Xu, J. Jang-Jaccard, T. Liu, F. Sabrina, and J. Kwak, "Improved bidirectional GAN-based approach for network intrusion detection using one-class classifier," Computers, vol. 11, no. 6, p. 85, 2022.
[34] S. M. Kasongo and Y. Sun, "Performance analysis of intrusion detection systems using a feature selection method on the UNSW-NB15 dataset," Journal of Big Data, vol. 7, pp. 1-20, 2020.
[35] S. Khanday, H. Fatima, and N. Rakesh, "Implementation of intrusion detection model for DDoS attacks in lightweight IoT networks," Expert Systems with Applications, vol. 215, 2023, Art. no. 119330.