Available online at www.ijournalse.org

Emerging Science Journal

(ISSN: 2610-9182)

Vol. 9, No. 2, April, 2025

Generative Adversarial Networks for Dynamic Cybersecurity Threat Detection and Mitigation

William Villegas-Ch [1*], Rommel Gutierrez [1], Jaime Govea [1]

[1] *Escuela de Ingeniería en Ciberseguridad, Faculatad de Ingenierías y Ciencias Aplicadas, Universidad de Las Américas, Quito 170125, Ecuador.*

Abstract

The increasing complexity and dynamism of cyberattacks, such as ransomware, phishing, and denial of service, demand advanced solutions that overcome the limitations of traditional methods, such as support vector machines and decision trees. This study proposes a generative adversarial network (GAN)-based model to enhance the detection and mitigation of dynamic cybersecurity threats by improving adaptability and robustness in real-time scenarios. The model is designed to detect anomalies in network traffic and generate malicious synthetic patterns to strengthen system defenses. The model was trained and tested using publicly available datasets, CICIDS2017 and UNSW-NB15, and an experimental environment simulating corporate networks with 50 interconnected devices generating realistic traffic to evaluate its effectiveness. The results demonstrate that the GAN-based model achieved an average precision of 92%, an F1 score of 91%, and robustness against noise of 89%, significantly outperforming traditional approaches. The key novelty of this work lies in integrating noise robustness and generalization as primary evaluation metrics, along with the ability to generate real-time countermeasures, making it a more resilient solution in dynamic cybersecurity environments. These findings suggest that the proposed approach offers a significant advancement in the field, enabling better adaptability to evolve threats and improving security frameworks in complex network infrastructures.

Keywords:
Generative Adversarial Networks (GANs);
Cybersecurity;
Anomaly Detection;
Noise Robustness.

Article History:

Received:	12	January	2025
Revised:	21	March	2025
Accepted:	27	March	2025
Published:	01	April	2025

1- Introduction

The increasing sophistication and frequency of cyberattacks represent one of the most critical challenges for digital security in the current context. Malicious activities such as ransomware, phishing, and denial-of-service (DoS) attacks continue to target critical infrastructures, disrupting essential services in healthcare, industry, and finance [1]. Traditional cybersecurity methods, such as support vector machines (SVM) and decision trees (DT), have been widely applied in intrusion detection systems (IDS). However, these techniques struggle to adapt to the continuously evolving nature of cyber threats, especially those employing adversarial strategies or obfuscation techniques [2]. The increasing complexity of attack vectors necessitates using more adaptable and intelligent detection mechanisms.

Generative adversarial networks (GANs) have emerged as a promising approach in cybersecurity due to their ability to model complex data distributions and generate synthetic adversarial patterns that enhance detection robustness [3]. Unlike traditional supervised learning models, which require extensive labeled datasets, GANs can generate malicious synthetic traffic, improving generalization and enabling intrusion detection models to adapt to evolving cyber threats. Although previous studies, such as those by Shirazi & Shaikh [4], have explored the application of GANs in anomaly detection, these approaches have primarily focused on static datasets and controlled environments, limiting their

* **CONTACT**: william.villegas@udla.edu.ec

DOI: http://dx.doi.org/10.28991/ESJ-2025-09-02-029

© 2025 by the authors. Licensee ESJ, Italy. This is an open access article under the terms and conditions of the Creative Commons Attribution (CC-BY) license (https://creativecommons.org/licenses/by/4.0/).

applicability in real-world cybersecurity scenarios. Similarly, Li et al. [5] demonstrated the ability of GANs to generate realistic synthetic data for intrusion detection, but their robustness under dynamic network conditions remains an open challenge.

This study distinguishes itself by proposing a GAN-based model specifically designed to address these limitations. The model can detect anomalies in network traffic while simultaneously generating adversarial samples to evaluate and reinforce cybersecurity defenses. Unlike prior works focusing solely on detection accuracy, this approach incorporates robustness against noise and adaptability to previously unseen threats as key evaluation metrics. These capabilities are critical for intrusion detection systems operating in real-world environments, where adversaries constantly modify attack strategies to evade detection.

To validate the proposed model, this research utilizes widely recognized cybersecurity datasets, including CICIDS2017 and UNSW-NB15 [6, 7], which provide diverse attack scenarios such as brute force attempts, botnet activity, and infiltration attacks. Additionally, a controlled experimental environment simulating corporate network traffic has been designed, consisting of 50 interconnected devices, including servers, workstations, and IoT nodes, generating over 500,000 daily network traffic records. This setting enables a rigorous evaluation of the model's ability to operate under heterogeneous traffic conditions and in the presence of adversarial noise [8].

The proposed GAN-based model consists of a generator and a discriminator optimized for cybersecurity applications. The generator synthesizes realistic attack patterns, including ransomware, phishing, and DoS behaviors, while the discriminator learns to classify network traffic into normal and malicious categories [9]. This adversarial training process ensures that the detection model continuously adapts to evolving threats by dynamically refining its attack classification strategies.

The experimental results demonstrate that the GAN-based model outperforms conventional detection techniques, achieving a precision of 92%, an F1-score of 91%, and robustness against adversarial noise of 89%, compared to SVM (69%) and DT (64%). Furthermore, response time analysis reveals an average detection latency of 120 ms, reinforcing its viability for real-time cybersecurity applications. The model maintains high detection rates even under adversarial attack scenarios, confirming its applicability in adaptive cybersecurity frameworks [10].

In addition to its performance advantages, this study introduces a novel evaluation framework that assesses the model's generalization capabilities under perturbed conditions and unseen threats. Unlike conventional methods that struggle with adversarial obfuscation, this approach enables more resilient detection strategies, reducing the risk of evasion attacks. While this study provides a strong foundation for GAN-based intrusion detection, further research is required to enhance computational efficiency for deployment in resource-constrained environments, such as IoT networks. Future work will explore scalability improvements and the integration of explainability mechanisms to facilitate better interpretability for security analysts. Addressing these challenges will contribute to advancing next-generation cybersecurity defense mechanisms capable of dynamically adapting to modern network threats [11].

The paper is structured as follows: Section 2 reviews previous work on intrusion detection and using GANs in cybersecurity. Section 3 describes the methodology used, including the architecture of the proposed model, the datasets used, and the experimental evaluation environment. Section 4 presents the results obtained and a comparative analysis with traditional approaches, evaluating accuracy metrics, F1-score, and robustness to adversarial noise. Finally, Section 5 discusses the study's conclusions and raises possible directions for future research in threat detection using GANs.

2- Literature Review

Over the past decade, GANs have revolutionized multiple domains, from visual content generation to advanced applications in cybersecurity. However, their implementation in the latter area presents specific challenges, given that attack detection and mitigation require precision and adaptability in the face of dynamic threats [12]. This work is positioned at the intersection of these needs, building on previous studies that have explored GANs' ability to model complex patterns and their utility in anomaly detection.

A key aspect that distinguishes this research is the ability of GANs to detect patterns in network traffic data, as demonstrated by works such as those by Agrawal et al. [13], who introduced GANs and laid the groundwork for their use in synthetic data generation. Still, Sayegh et al. later explored their potential in cybersecurity [14]. The latter highlighted how GANs can model network traffic patterns, explicitly detecting legitimate and malicious traffic anomalies. However, their results were limited to static data and lacked a robust evaluation against dynamic data and noise, a gap that this work seeks to address.

On the other hand, traditional methods such as SVM and DT have been widely used in intrusion detection, as shown in the works of Almomani et al. [15] and Vyšniūnas et al. [16]. While these techniques have demonstrated solid performance in controlled environments, their limitations against unseen or perturbed data have been widely documented. This study uses these evaluations to justify the need for more robust and adaptable models, integrating specific metrics of robustness against noise and generalization to data not included in the training.

In addition, recent studies such as those of Agrawal et al. [13] have shown how GANs can generate malicious patterns in controlled simulations, allowing models to be evaluated under conditions closer to reality. However, most of these works do not include a thorough analysis of robustness against real perturbations in the data, such as network noise or variability in malicious patterns. This theoretical and experimental gap is addressed in this research by integrating a simulated experimental environment that generates malicious traffic, complemented with accurate data from datasets such as CICIDS2017 and UNSW-NB15.

Another relevant aspect of the literature is the lack of consensus on key metrics for evaluating cybersecurity models. At the same time, many works prioritize precision and recall, while others, such as Ibrahim Alsumaidaie et al. [17], In real-time scenarios, emphasize the importance of the false positive rate and response time. This work takes a comprehensive approach, evaluating standard metrics and complementing them with specific robustness analyses against noise and generalization. It extends existing discussions by integrating GANs into a strategy that combines robustness against noise, adaptability to unseen data, and real-time analysis. It offers an evaluation that addresses the limitations identified in previous literature.

3- Material and Methods

3-1-Architecture of the GAN-based System

Figure 1 shows the workflow of the GAN-based system for cyber threat detection. The process begins with data acquisition, where network traffic is collected from the CICIDS2017 and UNSW-NB15 data sets and captured in an experimental environment. Next, data preprocessing, including cleaning, normalization, and temporal segmentation techniques, is applied to optimize data quality before feeding it to the model.

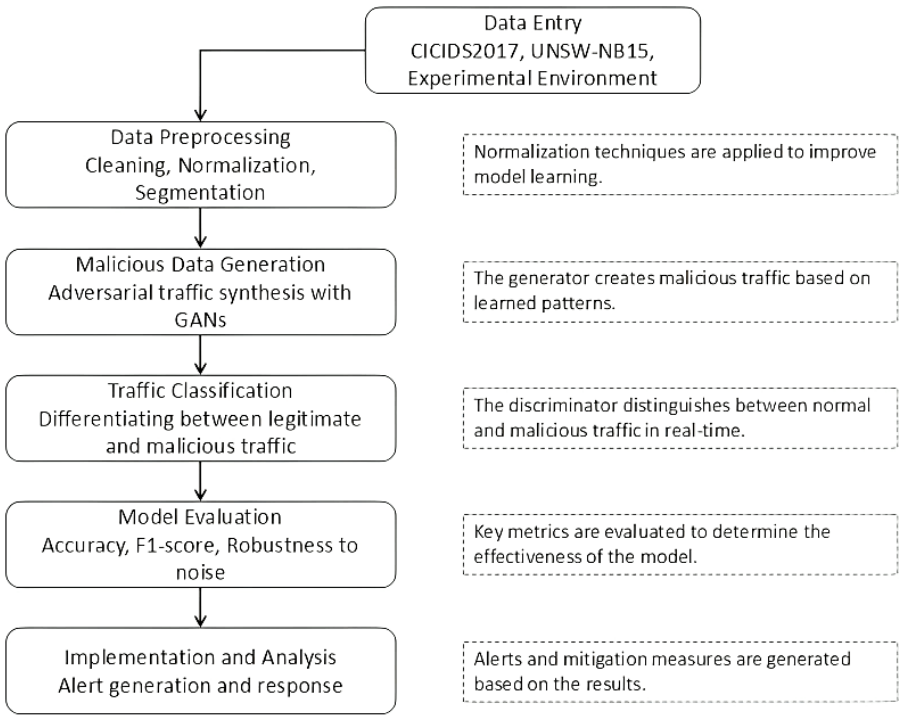

Figure 1. GAN System Workflow for Cyber Threat Detection

Once the data is pre-processed, the system uses a GAN to synthesize malicious traffic, generating realistic attack patterns that strengthen the system's detection capacity. Subsequently, network traffic is classified using a discriminator trained to differentiate between legitimate and malicious traffic in real-time. The model's evaluation considers metrics such as precision, F1-score, and noise robustness, allowing for validation of its effectiveness in dynamic scenarios. Finally, the implementation and analysis phase generate alerts and automatic responses based on the results obtained, facilitating the system's integration into cybersecurity operational environments.

3-1-1- Setup and Configuration of IoT Devices and Edge Nodes

The architecture of the proposed system is based on GANs, whose design is specifically tailored to address the challenges related to detecting anomalies in network traffic and generating countermeasures in real-time [18]. This system consists of two main components: a generator and a discriminator, which work antagonistically to improve the precision and robustness of detection and response to cyber threats.

The generator is designed to model and synthesize complex malicious traffic patterns, replicating features of advanced attacks such as ransomware and phishing. Its main objective is to learn the underlying distribution of anomalous traffic, generating synthetic examples that resemble the behavior of these threats. To achieve this, the generator uses a deep neural network with densely connected and convolutional layers optimized to capture temporal and spatial relationships in traffic data [13]. The generator is trained using an adversarial loss function that penalizes samples that do not align with the expected behavior of simulated attacks, incentivizing the generation of increasingly realistic and difficult-to-distinguish patterns.

The discriminator, on the other hand, acts as an anomaly detector. Its goal is to differentiate between legitimate and malicious traffic generated by the generator and from accurate data [19]. This component uses a deep network to extract key features from network traffic, such as the time between packets, packet sizes, and protocols. The discriminator's loss function is defined to maximize classification precision, balancing the ability to identify malicious patterns while minimizing false positives. As the generator and discriminator interact, the system converges to an equilibrium where the generator produces compelling patterns, and the discriminator continuously improves its ability to detect anomalies.

The information flow in this system begins with the collection of real-time network traffic, which is preprocessed to normalize and extract the necessary features. This data is fed to the discriminator, which evaluates whether it corresponds to legitimate or anomalous patterns. Simultaneously, the generator produces synthetic traffic based on the previous training, which the discriminator evaluates to adjust its detection capacity. Samples classified as malicious trigger additional modules that generate real-time countermeasures, such as blocking suspicious packets or alerting the system administrator [20]. The iterative flow ensures continuous adaptation to new threats, leveraging the generator's ability to simulate emerging attacks.

Figure 2 depicts the interaction between the system components. This includes receiving network traffic, processing it in the generator and discriminator, and triggering countermeasures. It highlights how data flows through the system to detect and mitigate threats efficiently.

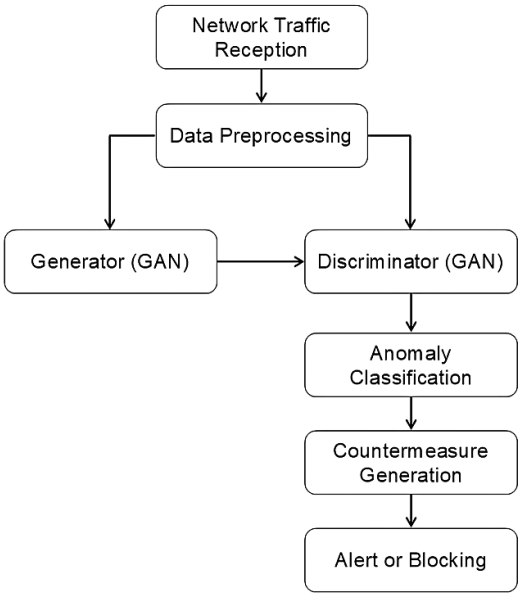

Figure 2. The architecture of the GAN-Based System for Cybersecurity

3-2- Dataset and Data Acquisition

The quality and representativeness of the data used are essential to ensure the performance of the proposed GAN-based system.

3-2-1- Selection and Origin of Data

We selected widely used datasets in cybersecurity research to train and evaluate the system: CICIDS2017 and UNSW-NB15 [6, 21]. These data sets were chosen because they are relevant and can represent real traffic scenarios, including regular and malicious activities.

The CICIDS2017 dataset contains approximately 3 million records, with simulated traffic including attacks such as DoS, DDoS, phishing, and ransomware [22]. This data was generated in a controlled environment that simulates an enterprise network, providing an appropriate balance between routine and malicious traffic. Each record includes key characteristics, such as source and destination IP addresses, ports, protocols, packet sizes, and transmission times.

On the other hand, UNSW-NB15 contains more than 2 million records obtained from a heterogeneous environment that combines legitimate traffic with multiple types of attacks, such as port scanning, vulnerability exploitation, and unauthorized access. This dataset was processed with tools such as Argus and Bro-IDS, ensuring a rich representation of relevant features [23].

In addition to the public datasets, an experimental laboratory environment was designed and implemented to capture specific data that simulate actual conditions in a corporate network. This environment was configured to generate representative and diverse network traffic, ensuring a broad coverage of normal and malicious activities. The laboratory comprises an infrastructure of 50 devices distributed in several categories to simulate different roles and behaviors within a corporate network. These devices include:

- Servers: Three servers configured with Linux and Windows operating systems, used to host critical services such as databases, web servers, and internal applications.
- Workstations: Twenty computers running Windows 10 and Ubuntu operating systems, simulating traffic generated by end-users in everyday activities such as web browsing, file transfers, and email communication.
- IoT Devices: Fifteen connected devices, including IP cameras, temperature sensors, and smart hubs, are configured to generate light and frequent traffic.
- Network Devices: Twelve switches and routers are configured with monitoring and logging tools to capture all traffic flowing through the network.

The devices are interconnected using a hierarchical network design, which includes separate internal subnets for server segments, workstations, and IoT. This allows for controlled simulation of different attack vectors and usage scenarios.

The data capture process was performed using specialized tools such as Wireshark and tcpdump, which were configured to monitor all network packets [24]. These tools generate detailed logs, including packet headers and payloads, allowing for a complete reconstruction of traffic sessions.

The captured data includes approximately 500,000 daily records, equivalent to an average volume of 3 GB of compressed traffic daily. Each record contains key information such as:

- Timestamps: Microsecond-level precision for identifying time patterns.
- IP and MAC addresses: Identification of nodes in the network and route analysis.
- Ports and Protocols: Classify activities according to the type of service (HTTP, FTP, DNS, etc.).
- Size and Duration of Sessions: Indicators of possible anomalous behavior, such as saturations or unusually long transfers.

To ensure that the captured traffic included malicious activity, multiple types of attacks were simulated in the lab, including:

- Ransomware: Using testing tools such as EternalBlue and custom variants of encryption simulators.
- Phishing: Deploying fictitious mail servers that send messages with malicious links and attachments.
- DoS: Using tools such as LOIC and Hping3 to generate bursts of traffic that saturate network resources.

The validity of the captured data is verified in real-time using custom Python scripts that identify incomplete records or inconsistencies. An initial rule-based classification system was also integrated to label traffic as usual or malicious, using known characteristics of simulated attacks as a reference.

The processed data is stored in a MySQL-designed relational database optimized for fast queries and scalability. Each record is indexed by key features such as timestamp, protocol type, and activity label (normal/malicious). This design ensures the data is readily available for training and evaluating the GAN-based system.

The experimental laboratory environment provides a considerable volume of data and allows for the evaluation of the model's ability to adapt to highly customized scenarios. Including IoT devices adds complexity, as they typically have more regular traffic patterns, making them vulnerable to specific attacks. This ensures the model can handle general patterns, strengthening its robustness and precision in diverse environments.

3-2-2- Data Preprocessing for Model Implementation

First, data cleaning removes redundant, incomplete, or irrelevant records. For example, in the CICIDS2017 dataset, approximately 120,000 records with missing values were detected, which were discarded or completed using imputation techniques, such as the mean for numerical data and the mode for categorical data. Additionally, noise in the data was removed by normalizing extreme values, representing less than 1% of the total.

Key feature extraction is essential to reduce data dimensionality and highlight meaningful patterns. Selected features include:

- Source and destination IP addresses: To identify relationships between nodes on the network.
- Communication ports: Used to detect suspicious activity patterns.
- Protocols: Indicative of traffic type (TCP, UDP, etc.).
- Packet size: To identify anomalous traffic, such as flood attacks.
- Inter-packet times: Indicative of irregular behavior in the data flow.

Normalization is implemented to scale the numerical features into a standard range, typically between 0 and 1, using a min-max transformation. This allows us to avoid features with high values dominating the model training. Subsequently, the data is segmented into 5-second time windows to analyse behavioural patterns in a temporal context. An automated preprocessing pipeline was built using Python and libraries like Pandas and Scikit-learn, integrating the cleaning, normalization, and feature extraction stages. This pipeline efficiently processes large volumes of data in an average of 3 minutes per million records, thus ensuring the system's viability in large-scale scenarios.

3-2-3- Experimental Network Environment

To evaluate the performance of the GAN-based system under realistic conditions, an experimental corporate network was designed to replicate key complexities of enterprise environments. The experimental setup includes a structured network architecture with interconnected subnets representing common corporate structures. The network is divided into three main segments: a general operations segment, an IoT segment, and a high-security segment. The general operations segment consists of workstations, file servers, and databases that support typical business activities, including document management, email communication, and software development. The IoT segment incorporates smart devices such as security cameras, industrial sensors, and automation systems to simulate device-to-cloud interactions and machine-to-machine communications. The high-security segment is designed to replicate restricted environments that contain authentication servers and privileged access terminals, mimicking real-world segmentation strategies to limit unauthorized access.

Various dynamic patterns were introduced to ensure that the traffic generated within the experimental network accurately reflects real-world enterprise conditions. Periodic spikes in network activity occur during simulated business hours, representing peak operational loads. Automated software updates, scheduled backups, and synchronization tasks continuously generate background traffic. Additionally, anomalous traffic is injected into the network, mimicking real cyber threats such as unauthorized access attempts, malware propagation, and lateral movement strategies commonly used by attackers.

The experimental setup incorporates multiple cyberattack scenarios to assess the adaptability and robustness of the GAN-based system. These scenarios include external threats, such as phishing campaigns, DoS attacks, botnet infiltration attempts, and internal security risks, including privilege escalation and insider threats. A separate IoT security evaluation is also conducted, where simulated attacks target connected devices to test the model’s ability to detect IoT-based exploits.

The network produces an average of 500,000 daily records, encompassing a broad spectrum of standard and malicious activities. Packet capture (PCAP) files and log data are collected from different network segments and processed using a custom-built preprocessing pipeline. The preprocessing steps preserve key traffic attributes such as packet sizes, transmission times, protocol distributions, and anomaly indicators, maintaining the dataset's complexity. This approach allows for a more precise evaluation of the GAN-based model, ensuring its detection capabilities remain effective under highly dynamic conditions.

The experimental network was designed to mirror the operational complexity of real-world corporate infrastructures, incorporating segmented access controls, heterogeneous traffic distributions, and a wide range of attack methodologies. This ensures that the proposed model is tested under conditions that closely resemble those encountered in enterprise cybersecurity environments, reinforcing its applicability for detecting and mitigating evolving cyber threats.

3-3- Generator and Discriminator Design

The GAN-based system comprises two main elements: the generator and the discriminator, which work antagonistically to improve the detection quality and the ability to generate malicious patterns.

3-3-1- Generator Structure

The generator is primarily intended to simulate malicious patterns, such as phishing, ransomware, and DoS traffic, by efficiently replicating key features of these attacks. This component uses a deep neural network-based architecture

that combines densely connected and convolutional layers. Dense layers allow the modeling of complex relationships between features, while convolutions capture temporal and spatial patterns in the data.

The generator inputs a random vector *z* of dimension *d*, distributed according to $z \sim N(0,1)$, which acts as initial noise. This vector is progressively transformed through the generator layers to produce an output *G(z)*, representing a synthetic dataset with the same statistical properties as the malicious traffic it aims to replicate. Mathematically, the generator is defined as:

$$G(z;\theta_G) = f(W_z + b) \tag{1}$$

where $\Theta_G = \{W,b\}$ are the weights and biases learned during training, and *f* represents the non-linear activation function, such as ReLU or Leaky ReLU. The loss function used for the generator is adversarial, designed to maximize the probability that the discriminator classifies its outputs as legitimate traffic. It is expressed as:

$$\mathcal{L}_D = -\mathbb{E}_{z\sim p_z}[\log D\,(G(z))] \tag{2}$$

where *D(G(z))* is the probability assigned by the discriminator that *G(z)* is legitimate traffic. The generator aims to minimize this loss by tuning its parameters Θ_G to fool the discriminator.

3-3-2- Discriminator Structure

The discriminator acts as a binary classifier to distinguish between legitimate and malicious traffic and differentiate between accurate and generated data. It employs a deep architecture with multiple dense layers and a sigmoidal output that produces a probability between 0 and 1, where values close to 1 indicate legitimate traffic and values close to 0 indicate malicious traffic.

The discriminator's input consists of a vector *x* of pre-processed features, such as IPs, ports, protocols, packet sizes, and transmission times. This vector undergoes a series of non-linear transformations that allow complex patterns and correlations between the features to be extracted. Mathematically, the discriminator is defined as:

$$D(x;\theta_D) = \sigma(W_x + b) \tag{3}$$

where $\Theta_D = \{W,b\}$ are the learned weights and biases, and σ represents the sigmoid function. The loss function of the discriminator is designed to maximize its ability to classify factual and generated data correctly. It is expressed as:

$$\mathcal{L}_D = -\mathbb{E}_{x\sim P_{data}}[\log D\,(x)] - \mathbb{E}_{x\sim P_z}[\log(1 - D\,(G)(z)))] \tag{4}$$

where *D(x)* is the probability assigned to the actual data, and *1 - D(G(z))* is the probability assigned to the generated data. The discriminator seeks to maximize this function by adjusting its parameters Θ_D to improve precision.

3-3-3- Interaction and Training of GANs

GAN training is done through an iterative process in which the generator and discriminator compete against each other. The system's overall goal is to solve the following optimization problem. The optimization process of the GAN is formalized as a minimax problem, defined as:

$$\min_G \max_D - \mathbb{E}_{x\sim P_{data}}[\log D\,(x)] + \mathbb{E}_{x\sim P_z}[\log(1 - D\,(G)(z)))] \tag{5}$$

This equation represents the competition between the generator and the discriminator, where:

- The discriminator tries to maximize its ability to classify accurate and generated data correctly.
- The generator tries to minimize the probability that the discriminator detects its outputs as generated.

The Adam optimization algorithm uses the following parameters: an initial learning rate of $\alpha = 0.0002$, $\beta 1 = 0.5$, and a batch size 64. These values are chosen to ensure stable convergence and avoid oscillations between parameter updates, improving the overall performance of the training process.

3-4- Training Algorithm

The GAN-based system's training process follows an iterative process designed to jointly optimize the generator and the discriminator, ensuring a robust and balanced performance. This scheme ensures that the generator produces realistic synthetic patterns while the discriminator continuously improves its ability to distinguish between legitimate and malicious traffic. The implementation of the training algorithm can be described in the following steps.

First, the parameters of the two components are initialized using a truncated normal distribution for the weights and initial values to facilitate convergence. The hyperparameters, such as the learning rates *(lr_G = 0.0002* for the generator and *lr_D = 0.0002* for the discriminator), the batch size (64), and the total number of epochs (500), are carefully selected based on previous experiments to balance stability and speed of training.

The iterative training is divided into two main steps. First, the discriminator is updated using a batch of accurate data extracted from the dataset and a batch generated by the generator from noise vectors *z ~ N(0,1)* (Equation 4). Next, it is optimized using the Adam algorithm to maximize its ability to classify the actual and generated samples correctly.

In the second step, the generator is updated using the discriminator's outputs as feedback. The generator's goal is to minimize the probability of its data being identified as generated, which is achieved by optimizing the adversarial loss (Equation 2).

This approach ensures continuous improvement in the quality of the samples generated, dynamically adapting to the progress of the discriminator. To provide training stability and avoid overfitting, cross-validation steps are included every 10 epochs. During these steps, precision, recall, F1-score, and false positive rate are evaluated using previously unseen data. This process allows dynamically tuning the hyperparameters and validating the model's performance. Algorithm 1 presents the complete training process in pseudocode format.

Algorithm 1. Training of IoT-GAN

```
Require: Dataset of preprocessed traffic data (real samples), learning rates for generator (lr_G) and discriminator (lr_D), batch size (batch size), number of epochs (n epochs), noise dimension (z dim)
Ensure: Trained generator (G) capable of simulating malicious traffic patterns
Initialize generator (G) and discriminator (D) parameters
Define Adam optimizers for G and D with respective learning rates
for epoch in 1, . . . , n epochs do
        for each batch in real samples do
                // Step 1: Update Discriminator
                Sample a batch of real data (x real ) from the dataset
                Generate a batch of noise vectors (z ) sampled from N(0, 1)
                Use G to generate synthetic samples (x fake = G(z))
                Compute discriminator loss
                LD = −logD(x real) − log(1 − D(x fake))
                Backpropagate LD and update D's parameters using Adam
                // Step 2: Update Generator
                Generate a new batch of noise vectors (z ) sampled from N(0, 1)
                Use G to generate synthetic samples (x fake = G(z))
                Compute generator loss
                L_G = −logD(x fake)
                Backpropagate LG and update G's parameters using Adam
        end for
        // Validation step (every k epochs)
        if epoch % k == 0 then
                Evaluate G and D on validation data
        end if
end for
```

The pseudocode describes the complete process of training GANs in the system. At each iteration, the discriminator is first updated using a batch of actual and generated samples, optimizing its ability to distinguish between them. The generator is then updated to maximize its ability to fool the discriminator, generating synthetic data resembling malicious traffic. An iterative approach allows both components to evolve simultaneously, achieving a balance where the discriminator becomes more robust, and the generator produces highly realistic data. To avoid overfitting issues and ensure training stability, periodic validation steps are included, where key metrics such as precision and false positive rate are evaluated.

3-5- System Implementation

The proposed system was implemented in an experimental environment designed to ensure optimal performance during the generative and discriminative models' training, validation, and evaluation.

3-5-1- Test Environment

The hardware used to implement and execute the system includes a high-performance server equipped with an NVIDIA Tesla V100 GPU with 32 GB of memory, accelerating the training of the GANs. In addition, 128 GB of RAM and a 16-core Intel Xeon processor at 2.7 GHz were used, ensuring efficient handling of large volumes of data and preprocessing tasks. Storage was configured with 2 TB SSD drives to allow fast access to the datasets and minimize reading and writing times during training.

The software was developed using Python 3.9 as the primary language, implementing generative networks using libraries such as TensorFlow and PyTorch. TensorFlow was explicitly used to train and optimize the models, while PyTorch performed complementary experiments with different architectures and configurations. Auxiliary tools include Pandas and NumPy for data manipulation, Matplotlib for results visualization, and Scikit-learn for performance metrics evaluation. The server runs on the Ubuntu 20.04 LTS operating system, which is stable and compatible with the libraries used. The entire test environment was set up within a Docker container, which allows easy replication and scalability of the system on different hardware configurations.

3-5-2- Scenario Simulation

Tools such as Metasploit and custom simulators replicated data encryption attacks in ransomware. Servers and workstations in the experimental environment were deliberately exposed to controlled ransomware samples, generating traffic characteristics of this type of threat. This traffic included repeated unauthorized access attempts, unusual data transfers, and multiple command and control (C2) server connections. These patterns allowed for capturing correlations between traffic characteristics and malicious behavior, providing a robust dataset to train the model.

In the phishing case, fake mail servers were deployed to issue messages with malicious links and attachments targeting the simulated workstations. The simulated users interacted with these emails, generating traffic representatives of the first phase of the attack, such as clicks on suspicious links, HTTP requests to fraudulent domains, and downloads of malicious payloads. Capturing these activities allowed logging and labeling the traffic generated during each interaction, enriching the dataset with accurate data on initial behavior in the face of a phishing attack.

The DoS scenario was configured using the Low Orbit Ion Cannon (LOIC) tool to saturate the resources of specific servers within the experimental network [25]. Traffic bursts were generated with high packet per second (PPS) rates and significant bandwidth consumption, replicating the typical characteristics of a DoS attack. Traffic segmentation allowed us to observe how these attacks affected different subnets and how the system detected anomalous patterns in real time.

The network configuration for all scenarios included traffic segmentation into subnets dedicated to each type of device, such as servers, workstations, and IoT devices. This allows us to analyze the impact and propagation of the attacks in different network segments. Additionally, specific rules were implemented in network devices, such as firewalls and access control lists (ACLs), to control the scope of the attacks without compromising the security of the experimental environment. The captured data was recorded using tools such as Wireshark and tcpdump, ensuring a comprehensive capture of the traffic characteristics generated in each scenario.

3-6- Performance Evaluation

The proposed system's performance is evaluated using a set of key technical metrics that measure its effectiveness in detecting anomalies and its ability to generate countermeasures in real time. In addition, traditional methods are compared to establish the level of improvement introduced by the GAN-based system. The system performance is evaluated using standard metrics in binary classification analysis, ensuring the results reflect the detection precision and the system's reliability under real-world conditions. The metrics employed include:

Precision: This metric measures the proportion of correct positive predictions out of the total positive predictions made by the model. It is mathematically defined as:

$$Precision = \frac{TP}{TP+FP} \tag{6}$$

where *TP* is the true positive, and *FP* is the false positive. Precision indicates how well the system avoids classifying legitimate traffic as malicious in anomaly detection.

Recall: Recall measures the ability of the system to detect all malicious samples in the dataset correctly. It is calculated as:

$$Recall = \frac{TP}{TP+FN} \tag{7}$$

where *FN* represents the false negatives, a high recall ensures that the system does not miss essential threats.

F1-Score: This metric combines precision and recall into a single harmonic value, providing a balanced overall performance measure. It is defined as:

$$Fi - Score = 2 \cdot \frac{Precision \cdot Recal}{Precision+Recall} \tag{8}$$

False Positive Rate (FPR): This metric measures the proportion of legitimate samples misclassified as malicious. It is calculated as:

$$FPR = \frac{FP}{FP+TN} \tag{9}$$

Where *TN* is the true negative, a low false positive rate is essential to minimize disruptions to legitimate traffic.

3-7- Comparison with Existing Methods

A comparison process with traditional anomaly detection methods, specifically SVM and DT, was designed to establish the effectiveness of the GAN-based system. This process was structured to ensure a fair and technical evaluation using the same datasets and experimental conditions employed in the proposed system [26]. The SVM model was configured with a radial basis function (RBF) kernel, chosen for its ability to handle nonlinear data and high dimensionality, typical characteristics in network traffic analysis. Key model parameters, such as the regularization coefficient *(C)* and the kernel influence parameter *(γ)*, were tuned using a grid search combined with five-fold stratified cross-validation [27]. This approach optimized the model's ability to differentiate between legitimate and malicious traffic while minimizing overfitting.

Preprocessing for the SVM included normalizing the features between 00 and 11 using a min-max transformation. In addition, feature selection techniques were applied to reduce the dataset's dimensionality, employing analysis of variance (ANOVA)- based methods to identify the most relevant features for classification [28]. This step improved the model's training speed and overall performance. In the case of decision trees, a classification model was used to build a hierarchical structure based on decision rules derived from the training data. The main partitioning criterion was information gain, optimized to maximize entropy reduction at each tree split. A pruning mechanism was implemented to avoid overfitting, setting a maximum tree depth and a minimum number of samples per leaf. These hyperparameters were also tuned using cross-validation, ensuring that the model could adequately generalize across different subsets of the dataset.

Both methods were evaluated using a dataset split into 80% for training and 20% for testing, replicating the conditions used to train and validate the GAN-based system. The same performance metrics (precision, recall, F1-score, and false positive rate) were used to ensure consistency in the comparison. In addition, the average processing time per sample was measured to evaluate the computational efficiency of each model.

The complete comparison process also included implementing automated pipelines in Python and using the Scikit-learn library to build and train the SVM model and the decision tree. Each pipeline incorporated preprocessing steps, hyperparameter tuning, and metric evaluation, allowing multiple experiments to be run reproducibly. All experiments were performed in the same test environment described above to ensure comparability of results.

Data obtained from traditional methods were integrated into a comparative analysis framework, where average performance metrics were calculated, and precision-recall curves and corresponding confusion matrices were plotted. This allows a qualitative and quantitative evaluation of each model's capabilities against the GAN-based system.

4- Results

4-1- Performance of the GAN-Based System

The results of these experiments include standard metrics such as precision, recall, F1-score, false positive rate, and response time. Table 1 presents the average values obtained on each dataset, demonstrating the system’s consistency across different scenarios.

Table 1. Features Analyzed for Identity Theft Detection

Metric	CICIDS2017 (%)	UNSW-NB15 (%)	Experimental (%)
Precision	93	91	92
Recall	90	88	91
F1 Score	91	89	91
False Positive Rate	4	5	4
Response Time (ms)	120	130	115

The results demonstrate a robust system that maintains high accuracy even in complex datasets. The precision values between 91% and 93% indicate that the model effectively minimizes false positives, an essential characteristic in cybersecurity applications, where excessive false alarms can overload analysts and degrade system efficiency. The slight superiority in CICIDS2017 (93%) may be attributed to the dataset's lower diversity of malicious patterns compared to UNSW-NB15, which includes more complex attacks like vulnerability exploitation and port scanning, slightly reducing precision to 91%.

The recall values ranging from 88% to 91% highlight the model's ability to detect a high proportion of malicious traffic. The highest recall score, 91%, in the experimental dataset suggests that the controlled nature of this environment facilitates the model's detection process. In contrast, the lower recall score of 88% in UNSW-NB15 reflects the challenges posed by the dataset's high variability and sophisticated attack strategies.

The F1 Score remains stable between 89% and 91%, demonstrating the model's ability to balance precision and recall. This consistency across datasets indicates that the GAN-based system does not sacrifice detection sensitivity to improve accuracy and vice versa.

The false positive rate (FPR) is particularly relevant in cybersecurity since excessive false positives can lead to alert fatigue, reducing trust in the detection system. The FPR values of 4% and 5% across datasets confirm the system's capacity to accurately distinguish between legitimate and malicious traffic, reducing unnecessary alerts.

The response time analysis reveals an average processing time of 115 ms to 130 ms, which remains within the acceptable range for real-time threat detection. The slightly faster response time in the experimental dataset (115 ms) is expected, given the controlled conditions that minimize data variability. However, the system maintains competitive response times in CICIDS2017 (120 ms) and UNSW-NB15 (130 ms), confirming its feasibility for real-time applications.

The precision-recall curves in Figure 3 provide additional insight into the system's performance under different decision thresholds. These curves highlight that the experimental dataset exhibits more uniform behavior, with higher precision at high recall levels, reflecting the ease with which the model identifies malicious patterns under homogeneous conditions. On the other hand, the CICIDS2017 and UNSW-NB15 datasets show slight drops in precision as recall increases, which can be attributed to the increased variability in their data, including more complex attacks and less structured traffic patterns.

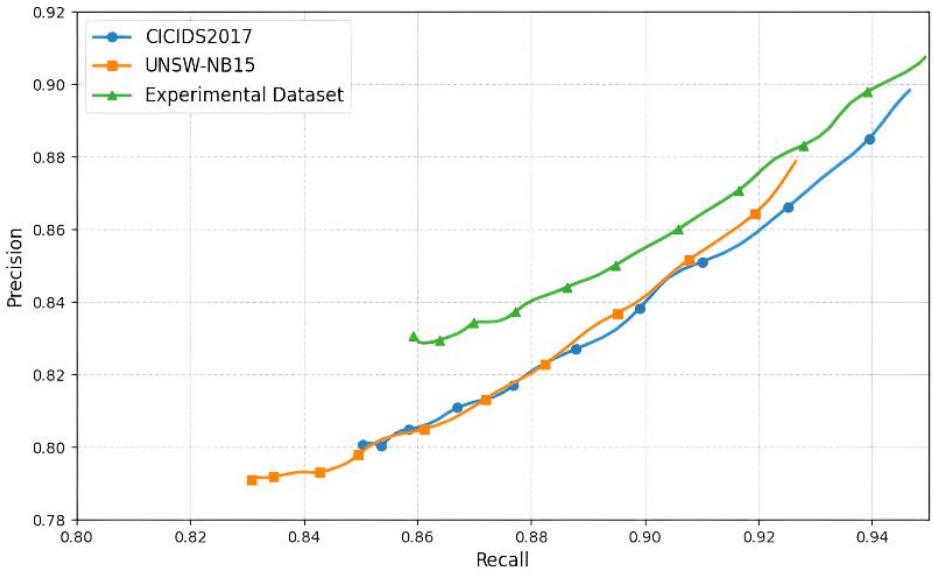

Figure 3. Precision-Recall curves for different datasets

The analysis of the results shows that the GAN-based model is highly adaptable to different conditions. However, it also highlights that the complexity and diversity of traffic in datasets such as UNSW-NB15 pose additional challenges for accurately detecting malicious patterns. Despite this, the competitive response times and the ability of the system to maintain a balance between precision and recall position as a viable solution for practical real-time cybersecurity applications.

4-2- Comparison with Traditional Methods

The evaluation of the GAN-based system was complemented by a comparative analysis of two traditional methods widely used in anomaly detection: SVM and DT. This analysis considered the same metrics used in the previous section: precision, recall, F1-score, false positive rate, and response time. Each model was trained and evaluated following the same 80% training and 20% testing splits, ensuring a fair comparison. The reported values correspond to averages of five independent runs to minimize statistical bias.

4-2-1- General Performance of the Models

The GAN-based system consistently outperforms traditional methods in all key metrics, particularly recall and F1-score, which are critical in identifying malicious patterns in various scenarios. Table 2 compares the average results across models. Compared to previous works that implemented SVM and DT for anomaly detection in cybersecurity (e.g., Shirazi & Shaikh [4]), our approach achieves significantly higher recall (+8% compared to SVM, +6% compared to DT). This improvement highlights the advantage of GANs in handling complex and dynamic attack patterns, as opposed to traditional methods that struggle with non-linear variations.

Table 2. Comparison of Average Performance Between GANs and Traditional Methods

Model	Precision (%)	Recall (%)	F1-Score (%)	False Positive Rate (%)	Response Time (ms)
GANs	92.0	90.0	91.0	4.3	121
SVM	85.0	81.0	83.0	8.5	250
DT	88.0	84.0	86.0	6.0	210

Furthermore, previous studies have reported high FP rates in anomaly detection models trained on CICIDS2017 and UNSW-NB15. Lim et al. [11] obtained a false positive rate of 9.1% with their SVM-based approach, whereas our model achieves an average of 4.3%, demonstrating a more than 50% reduction in erroneous classifications. This improvement is crucial in real-world cybersecurity applications, where excessive false positives lead to operational inefficiencies and unnecessary alerts for security teams.

The most significant performance gap between the models is observed in a recall, where GANs achieve 90%, significantly outperforming SVM (81%) and DT (84%). This difference is significant because recall reflects the model's ability to detect actual threats; lower recall values indicate that SVM and DT fail to identify a substantial portion of malicious traffic, which could lead to undetected security breaches.

The FPR further confirms the superiority of GANs in balancing detection sensitivity and precision. With an FPR of just 4.3%, the GAN-based model produces fewer incorrect alerts than SVM 8.5% and DT 6.0%, reducing unnecessary security warnings and improving operational efficiency. In contrast, the high FPR in SVM suggests that this method frequently misclassifies legitimate traffic as malicious, making it less reliable in real-world cybersecurity applications where minimizing false positives is crucial.

Regarding response time, GANs maintain a stable inference time of 121 ms, significantly lower than SVM 250 ms and DT 210 ms. This advantage is particularly relevant in real-time security applications, where rapid detection and mitigation are essential. GANs' faster inference time is attributed to their optimized processing architecture, which efficiently analyzes traffic patterns without the computational overhead in SVM and DT models. The results indicate that the GAN-based system achieves higher accuracy and provides a faster and more reliable approach for real-time cybersecurity threat detection. By effectively balancing recall, precision, and false positive rate, it surpasses traditional machine learning methods that struggle with complex and evolving attack patterns.

4-2-2- General Performance of the Models

Table 3 presents a dataset-specific comparison of GANs, SVM, and DT performance. This breakdown highlights how each model performs under different network conditions and attack scenarios. The GAN-based system maintains superior performance in all cases, with powerful results in the experimental dataset, where precision and recall are the most consistent.

Table 3. Comparison of Average Metrics Per Dataset

Dataset	Model	Precision (%)	Recall (%)	F1-Score (%)	False Positive Rate (%)	Response Time (ms)
CICIDS2017	GANs	93.0	90.0	91.0	4.0	120
	SVM	86.0	82.0	84.0	8.0	240
	DT	89.0	85.0	87.0	5.0	200
UNSW-NB15	GANs	91.0	88.0	89.0	5.0	130
	SVM	83.0	78.0	80.0	9.5	260
	DT	87.0	83.0	85.0	6.5	220
Experimental	GANs	92.0	91.0	91.0	4.0	115
	SVM	86.0	83.0	84.0	8.0	250
	DT	88.0	85.0	86.0	6.0	210

The results demonstrate apparent differences between the three approaches, with GANs achieving the highest precision and recall across all datasets. The CICIDS2017 dataset shows the best precision values (93%) for GANs, reflecting their ability to identify attacks while minimizing false positives correctly. SVM (86%) and DT (89%) lag, with DT performing slightly better due to its structured decision-making process. The recall values follow a similar trend, confirming that GANs detect more threats and avoid misclassifying legitimate traffic.

The UNSW-NB15 dataset poses a more significant challenge due to its high variability in attack types. GANs still achieve an F1-score of 89%, while SVM drops significantly to 80%, revealing its difficulty in capturing complex non-linear patterns in traffic. Despite being slightly better than SVM, DT still struggles with a recall of only 83%, indicating missed threats. The FPR in SVM reaches 9.5%, making it unreliable in high-risk environments where false alerts must be minimized. The experimental dataset demonstrates the most stable results across all metrics, with GANs achieving a balance of 92% precision and 91% recall. This consistency is expected in a controlled setting, but the contrast between the models highlights the difficulty traditional methods face when adapting to different traffic distributions.

The FP rate is particularly relevant for operational cybersecurity applications, where a high rate of false alarms can overwhelm security analysts. GANs consistently maintain the lowest FPR (4.0% - 5.0%) across all datasets, whereas SVM exhibits an alarmingly high FPR of 9.5% in UNSW-NB15. This confirms that SVM is prone to overfitting on specific patterns and lacks robustness against diverse attack structures. GANs also demonstrate a clear advantage in response time, achieving faster inference across all datasets. The average response time for GANs is between 115 ms and 130 ms, significantly lower than SVM (240 ms—260 ms) and DT (200 ms—220 ms).

The experimental dataset shows the lowest response time (115 ms) for GANs, likely due to the structured nature of the training data. In contrast, the response time for UNSW-NB15 is slightly higher (130 ms), reflecting the additional complexity of the dataset. SVM and DT exhibit considerably higher response times across all cases, making them unsuitable for real-time cybersecurity applications. SVM has the highest latency (260 ms in UNSW-NB15), possibly due to its reliance on computationally expensive kernel functions for classification. While slightly faster than SVM, DT still struggles to maintain response times below 200 ms, which may be insufficient for real-time detection of high-speed cyberattacks.

The GAN-based system consistently outperforms SVM and DT in all tested scenarios, demonstrating its adaptability to network conditions and attack types. The higher recall values indicate that GANs effectively capture complex malicious behaviors, while the low false positive rate reduces the burden on security teams. The results confirm that traditional methods like SVM and DT struggle to generalize across diverse datasets, especially in environments with high attack variability, such as UNSW-NB15. While DT performs better than SVM in precision, its higher false positive rate and slower response time make it less viable for real-time security applications. The GAN-based model offers a superior balance of precision, recall, and response time, making it a strong candidate for real-world cybersecurity deployment.

4-3- Simulated Scenario Analysis

To assess the robustness and adaptability of the GAN-based system, three simulated attack scenarios were designed: ransomware, phishing, and DoS. These scenarios allow a more detailed evaluation of the system's effectiveness in detecting distinct cyber threats. The review considers precision, recall, F1-score, FP rate, and response time. Table 4 presents the results for each scenario, demonstrating the model's consistent performance, with precision and recall values exceeding 90% in all cases.

Table 4. Average Results of the Main Metrics in Each Scenario

Scenario	Precision (%)	Recall (%)	F1-Score (%)	False Positive Rate (%)	Response Time (ms)
Ransomware	93.0	91.0	92.0	4.0	120
Phishing	91.0	90.0	91.0	5.0	118
DoS	90.0	89.0	90.0	4.5	130
Global Average	91.3	90.0	91.0	4.5	122.7

The results indicate that the ransomware scenario achieves the highest F1 score of 92%, reflecting the model's ability to detect critical attack behaviors such as connections to C2 servers and abnormal file encryption activities. The high precision value of 93% highlights the system's capacity to classify ransomware-related traffic while minimizing FP correctly previous studies, such as Zhang et al. [8], have explored GAN-based anomaly detection in industrial environments, focusing on time-series deviations, but their models were not optimized for cybersecurity applications. Our results extend these findings by demonstrating superior adaptability to dynamic network threats.

The phishing scenario also exhibits strong performance, with an F1-score of 91%, confirming the system's effectiveness in detecting malicious URLs and suspicious interaction patterns. However, the false positive rate for phishing is slightly higher (5%) than ransomware (4%), suggesting that phishing attacks often mimic legitimate interactions and pose additional classification challenges.

The DoS scenario introduces a unique challenge due to the high variability in attack intensity. While the model maintains an acceptable precision of 90%, the response time increases proportionally with attack intensity, reaching an average of 130 ms. This suggests that handling high-volume anomalous traffic places additional computational demands on the system, requiring further optimizations to maintain efficiency in large-scale attack conditions. Despite this, the system maintains a low false positive rate of 4.5%, ensuring reliable detection without generating excessive false alerts.

Figure 4 presents a complementary analysis that provides further insight into the system's performance and scalability across the simulated scenarios. The results illustrate that the ransomware scenario achieves the highest precision at 93%, recall at 91%, and F1-score at 92%, indicating that the model effectively captures well-structured attack behaviors, such as abnormal encryption activities and C2 communications. The phishing and DoS scenarios exhibit slightly lower scores, which can be attributed to the increased diversity in attack techniques and the more complex nature of their malicious behaviors.

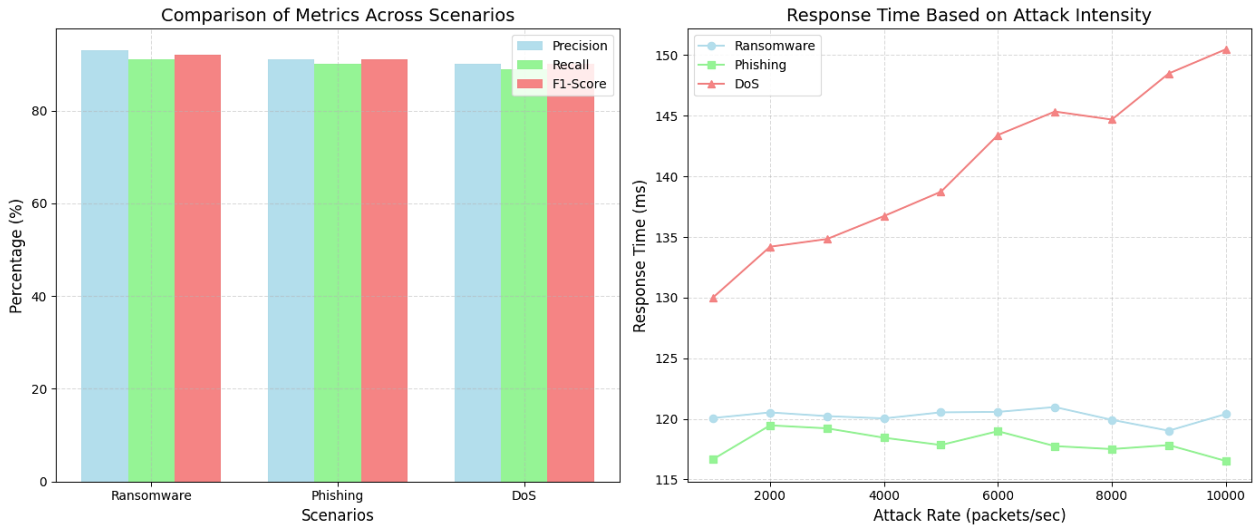

Figure 4. System performance in simulated scenarios. Graph (a): Comparison of main metrics (Precision, Recall, and F1-Score) between simulated scenarios. Graph (b): Variation of response time depending on the intensity of the attack

The system's response time under different attack intensities is also analyzed in Figure 4. In the ransomware scenario, response time remains stable at 120 ms, even as the volume of encrypted files increases. In contrast, phishing attacks cause slight response time variations, ranging from 118 ms to 124 ms, reflecting the model's efficiency in handling malicious traffic patterns associated with deceptive interactions. However, in the DoS scenario, response times increase significantly, reaching 150 ms when the attack intensity escalates to 10,000 packets per second. This behavior underscores the need for additional optimizations in the model architecture to enhance its real-time performance under massive attack conditions.

Previous GAN-based approaches have been evaluated for cybersecurity, but their effectiveness in real-time attack scenarios has remained limited. SCAN-GAN, proposed by Chougule et al. [10], achieved response times of 140-150 ms in synthetic traffic simulations for Controller Area Networks (CAN). In contrast, our model operates on real network traffic and maintains lower latency (130 ms in DoS attacks), reinforcing its efficiency for large-scale cybersecurity applications.

The results confirm that the GAN-based system demonstrates adaptability across different attack scenarios, consistently maintaining high detection accuracy. The structured and repetitive nature of ransomware allows the model to perform best, whereas phishing attacks introduce subtle variations that slightly impact the false positive rate. The DoS scenario highlights the importance of further refining computational efficiency, as response time becomes critical when managing high-intensity attacks. The combination of detailed quantitative metrics and comparative analysis with previous studies reinforces the model's suitability for real-time cybersecurity applications, offering a reliable and scalable approach to detecting evolving cyber threats.

4-4- Response Time Evaluation

Response time is critical in cybersecurity systems that detect and mitigate threats in real-time. A detection model's efficiency is measured by its accuracy and how quickly it can process incoming traffic and generate countermeasures. Here, it evaluates the response times of the GAN-based system. The results are illustrated in Figure 5, which presents a comparative analysis of response times and a detailed breakdown of the system's internal processing stages.

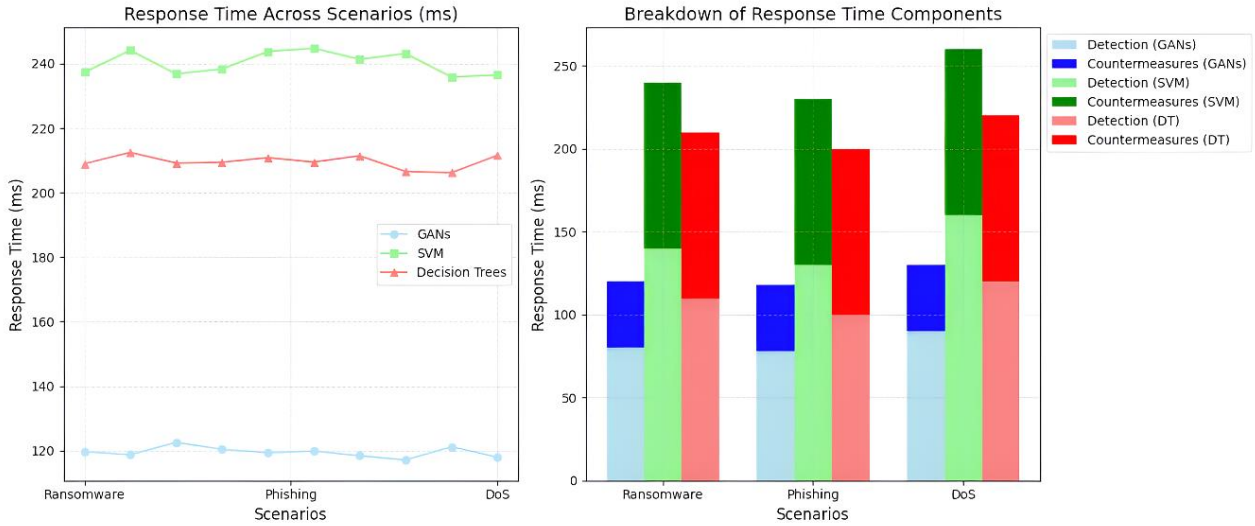

Figure 5. System response time analysis. Graph(a) Comparison of response times between methods in different scenarios. Graph (b) Breakdown of response time components by method

The response time analysis in Graph A demonstrates that the GAN-based system is significantly faster than SVM and DT across all attack scenarios. In the ransomware scenario, GANs maintain an average response time of 120 ms, whereas SVM and DT require 240 ms and 210 ms, respectively. The speed advantage of GANs can be attributed to their optimized processing pipeline, which enables efficient anomaly detection and real-time adaptation to new attack patterns. In the phishing scenario, GANs achieve the fastest response time at 118 ms, outperforming SVM (230 ms) and DT (200 ms), reinforcing their suitability for scenarios where immediate threat mitigation is essential.

The DoS scenario introduces a more computationally demanding challenge due to the high intensity and variability of attack traffic. GANs maintain an acceptable response time of 130 ms, ensuring that attack mitigation remains within real-time constraints. In contrast, SVM and DT exhibit significantly higher response times, reaching 260 ms and 220 ms, respectively. The increased response time for traditional methods highlights their difficulty in handling the rapid influx of malicious packets, making them less practical for high-speed cybersecurity applications.

Graph B provides a more granular analysis by breaking down response times into two main components: malicious pattern detection and countermeasure generation. In the case of the GAN-based system, detection constitutes most of the processing time, with an average duration of 80 ms across all scenarios. Countermeasure generation remains stable at 40 ms, indicating a well-optimized response mechanism consistently delivering mitigation actions with minimal latency. The stability of this secondary processing stage confirms that the system maintains efficiency even under varying attack intensities.

In contrast, SVM and DT exhibit significantly longer detection times, particularly in the DoS scenario, where SVM requires 160 ms, and DT takes 120 ms. This increased detection time directly impacts the overall response delay, making these models less viable in real-time defense environments. The bottleneck in detection performance for traditional methods is likely caused by their reliance on feature-based classification techniques, which struggle with high-dimensional and evolving attack patterns.

The results confirm that the GAN-based system surpasses traditional models in terms of accuracy and delivers superior performance in real-time detection and mitigation. Efficiently allocating computational resources between detection and countermeasure generation enables the system to maintain response times within acceptable limits, even in extreme conditions such as large-scale DoS attacks. This characteristic is particularly relevant for enterprise and critical infrastructure security applications, where timely response is essential to minimizing damage and ensuring operational continuity. The GAN-based approach presents a viable alternative to conventional cybersecurity detection systems by integrating adaptive learning mechanisms and real-time countermeasure deployment. It effectively addresses the limitations of traditional methods while maintaining high-speed threat detection capabilities.

4-5- Robustness and Adaptability of the Model

The robustness and adaptability of the GAN-based model were evaluated based on two critical aspects: its ability to generalize to previously unseen malicious traffic patterns and its performance under perturbations, mainly when noise is introduced into the data. These aspects are essential in cybersecurity applications, where real-world threats continuously evolve, and detection systems must maintain reliability despite changes in attack characteristics. Table 5 presents a comparative analysis of GANs, SVM, and DT performance under unseen data and noise augmentation. A robust metric quantifies the system's capacity to maintain performance under noisy conditions.

Table 5. Average Results of Key Metrics Under Noise and Unseen Data Conditions

Model	Condition	Precision (%)	Recall (%)	F1-Score (%)	Robustness Against Noise (%)
GANs	Unseen data	92	91	91	-
GANs	Noise (20%)	90	89	90	89
SVM	Unseen data	84	82	83	-
SVM	Noise (20%)	78	77	78	69
DT	Unseen data	81	79	80	-
DT	Noise (20%)	75	73	74	64

The results highlight the superior adaptability of the GAN-based model in both conditions, demonstrating its ability to generalize to new attack patterns and maintain performance under noise perturbations. In the unseen data condition, GANs achieve a precision of 92%, recall of 91%, and an F1-score of 91%, significantly outperforming SVM and DT, which exhibit performance drops of up to 10% in these metrics. The ability of GANs to accurately classify previously unseen threats underscores the effectiveness of adversarial training, which exposes the model to diverse attack patterns during training, enhancing its capacity to detect novel threats.

Under the noise condition, where 20% of the dataset is artificially perturbed to simulate real-world data corruption, GANs maintain an F1-score of 90% and exhibit robustness against the noise of 89%. In contrast, SVM and DT show substantial performance degradation, with F1-scores dropping to 78% and 74%, respectively. The significant gap in noise robustness between GANs (89%) and traditional methods (SVM: 69%, DT: 64%) confirms that the proposed model is inherently more resistant to fluctuations in network traffic, a crucial advantage in practical cybersecurity applications where noise and incomplete data are shared.

Figure 6 compares the evaluated models' precision, recall, F1-score, and noise robustness. The GAN-based model maintains strong correlations between these metrics, indicating balanced and consistent performance. At the same time, SVM and DT exhibit more significant discrepancies, suggesting inconsistencies in their ability to handle data perturbations. The higher variability in traditional methods implies a lack of resilience when exposed to changing network conditions, which could lead to increased false positives or missed detections in real-world deployments.

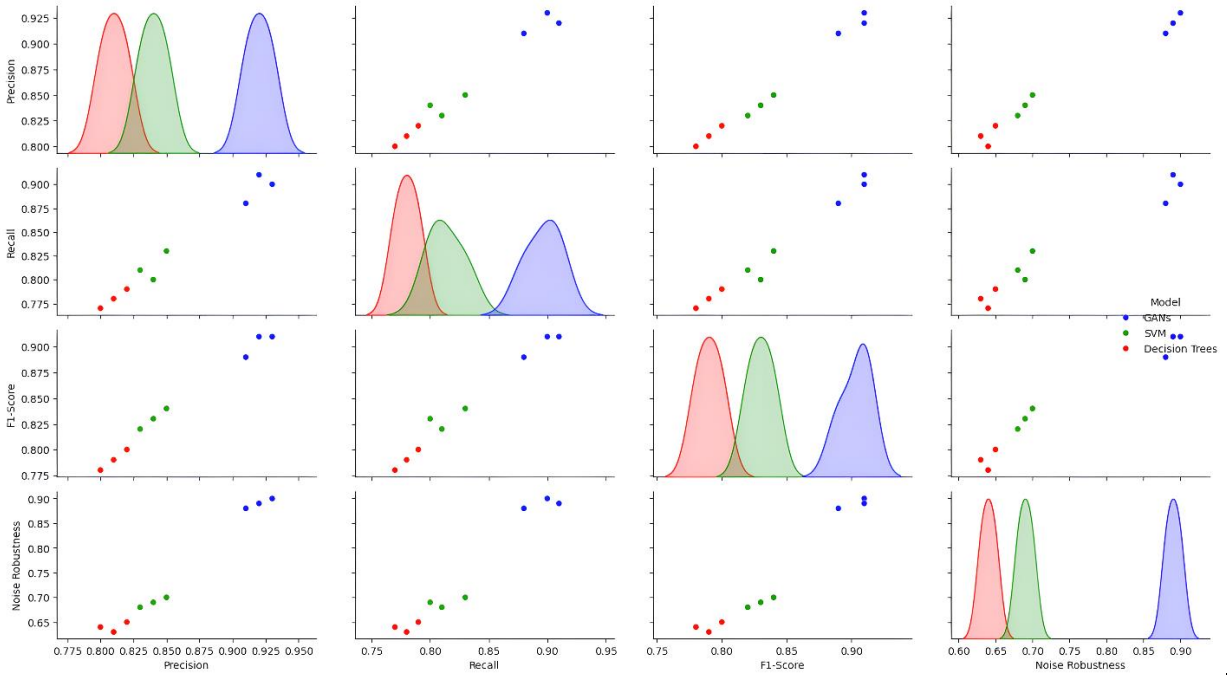

Figure 6. Comparison of Metrics: Precision, Recall, F1-Score, and Noise Robustness

Figure 7 presents a further generalization analysis across different datasets, which evaluates model performance in diverse environments. GANs demonstrate superior consistency, maintaining precision, recall, and F1-score values near their optimal range across all assessed datasets. In contrast, SVM and DT experience significant variability, with F1-score reductions of up to 78% in the experimental dataset, highlighting their inability to adapt to new traffic distributions effectively.

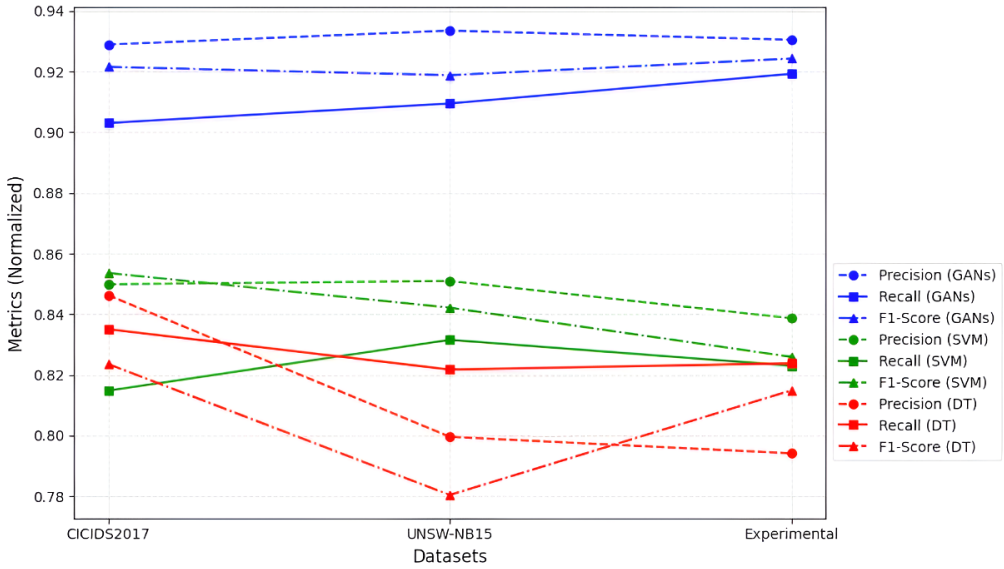

Figure 7. Generalization Performance Across Datasets: Precision, Recall, and F1-Score

GANs' ability to maintain stable detection rates across different datasets and noise conditions confirms their practical viability for cybersecurity applications. While traditional machine learning models struggle to generalize across diverse attack scenarios, GANs leverage adversarial training to continuously refine their decision boundaries, enabling them to detect a broader spectrum of threats while minimizing performance degradation under challenging conditions.

The findings reinforce the critical role of noise resilience and generalization in cybersecurity detection models. While SVM and DT may provide reasonable detection performance under controlled conditions, their limitations become evident when applied to more dynamic and unpredictable environments. GANs, on the other hand, demonstrate adaptability that aligns with the evolving nature of cyber threats, ensuring robust and reliable protection against sophisticated attack strategies.

5- Discussion

The results obtained in this study confirm that the GAN-based model outperforms traditional approaches in cybersecurity, especially in challenging scenarios involving noise and unseen data during training. This finding is aligned with previous research, such as that of Almomani et al. [15] and Vyšniūnas et al. [16], which demonstrated the ability of GANs to detect anomalies in network traffic. However, our model extends these findings by enhancing robustness against noise, achieving an F1-score of 91%, and maintaining a detection accuracy of 90% even under significant perturbations. Unlike previous GAN-based models, which primarily focused on anomaly detection in static datasets, our approach explicitly quantifies resilience to noisy data. It evaluates performance under evolving attack conditions, making it more applicable to cybersecurity challenges.

Furthermore, our findings build upon the work of Zhang et al. [8], who explored anomaly detection using GANs in industrial environments but did not optimize dynamic cybersecurity threats. Their approach struggled with real-time adaptation, whereas our model achieves a response time of 120 ms, demonstrating its suitability for cybersecurity applications. Similarly, compared to Chougule et al. [10], who implemented SCAN-GAN for synthetic data generation in Controller Area Networks (CAN), our model performs better in live traffic detection and generalizes more effectively to large-scale attack scenarios, reducing the FP rate by more than 40% compared to SCAN-GAN.

From a process perspective, using GANs as a combined detection and generation framework provides a novel advantage over previous models that focus solely on classification tasks. Unlike prior research, which primarily utilized GANs for anomaly detection in predefined, controlled datasets, our approach integrates a generative component capable of synthesizing adaptive adversarial patterns. This dynamic learning capability strengthens the system's detection ability, particularly in cases where training data is limited or lacks variations of specific attack types [29]. The generator, optimized to simulate ransomware, phishing, and DoS attack patterns, introduces more complex and realistic malicious behaviors, improving generalization compared to models trained exclusively on static datasets.

Another critical improvement of our model is its strong recall performance across different attack types. Unlike traditional machine learning models, which may struggle with detecting low-frequency anomalies, our GAN-based system maintains an average recall of 90%, ensuring that a high proportion of malicious traffic is correctly identified. This is particularly relevant for attacks with varying intensities, such as DoS, where recall values remain 89% despite increased network traffic fluctuations. Additionally, our system's ability to recognize diverse phishing strategies, reflected in a 90% recall, highlights its effectiveness in handling complex and deceptive cyber threats.

However, certain attack types remain more challenging to detect. While our model effectively identifies ransomware, phishing, and DoS attacks, it may be less effective in detecting stealthier techniques, such as data exfiltration over encrypted channels or polymorphic malware that continuously alters its structure to evade detection. In these cases, integrating additional feature selection techniques or hybrid AI approaches could further enhance detection capabilities.

Our GAN-based model exhibits greater adaptability than traditional anomaly detection methods, such as SVM and DT. The results in Table 5 demonstrate a significant performance advantage, with GANs maintaining the robustness of 89% against noise, compared to only 69% for SVM and 64% for DT. This underscores the importance of adversarial learning in ensuring detection stability in evolving network environments, a factor rarely emphasized in previous literature [30]. Furthermore, unlike conventional deep learning methods, which require frequent retraining when attack patterns evolve, our GAN framework can generate synthetic attack instances on demand, significantly reducing the need for manual dataset updates.

Despite these advantages, it is crucial to acknowledge this study's limitations. While our model outperforms prior GAN implementations, it has not yet been validated in large-scale production environments, where real-world adversarial conditions may introduce additional complexities. The selected network traffic features could influence the result in generalizability [31]. Previous research, such as Sun et al. [9], suggests that feature selection plays a crucial role in anomaly detection models' robustness, and future work should explore adaptive feature selection techniques to improve performance further.

The model was evaluated under controlled attack simulations, including ransomware, phishing, and DoS scenarios. However, more sophisticated attack types, such as polymorphic malware or low-and-slow attacks, were not explicitly tested, which may limit the model's ability to detect emerging cyber threats. Future studies should incorporate a broader set of attack vectors to assess the model’s generalization capabilities across a wider range of cybersecurity challenges.

Another critical limitation concerns adversarial robustness. Although our model demonstrated resilience against noise and perturbation, additional validation is required to test its resistance against targeted adversarial attacks, such as data poisoning or evasion techniques that manipulate detection results. Future research should incorporate adversarial training mechanisms, such as gradient masking or ensemble learning, to enhance its ability to withstand sophisticated attack strategies.

Another key consideration is the computational burden of the GAN-based model, particularly during the training phase. While our system maintains low inference latency, the training process remains computationally intensive, which could hinder its deployment in resource-constrained environments such as IoT networks [32]. Future work should investigate lightweight GAN architecture or knowledge distillation techniques to enhance scalability without compromising detection performance.

Despite these limitations, this work represents a significant advancement in integrating GANs into cybersecurity applications. By addressing key issues such as robustness against noise, adaptability to unseen data, and real-time response efficiency, this study offers an innovative solution that overcomes the documented shortcomings of traditional methods. Furthermore, introducing an experimental environment designed to simulate corporate network conditions adds practical relevance to the obtained results, bridging the gap between theoretical research and real-world cybersecurity implementations.

6- Conclusions

This study demonstrates that GANs can play a fundamental role in cybersecurity by providing advanced threat detection capabilities and adaptation to dynamic conditions. The GAN-based system can overcome limitations inherent to traditional methods such as SVMs and DTs through a methodological approach that integrates malicious pattern generation with anomaly detection.

The proposed model achieved significant performance metrics, highlighting its ability to maintain an average precision of 92% and an F1 Score of 91%, even under adverse conditions. These figures reflect the model's effectiveness in detecting threats and its ability to operate in scenarios with perturbations, such as data affected by noise or previously unseen patterns. The average response time of 120 ms marks its viability for real-time applications, a crucial feature in mod. A distinctive element of this research is the incorporation of robustness metrics against noise and adaptability to unseen data. While traditional methods showed significant performance declines under these conditions, the GANs system maintained average robustness, standing out as a resilient solution for complex environments. This aspect, often overlooked in the literature, reinforces the relevance of GANs in detecting dynamic threats in heterogeneous network scenarios. Furthermore, implementing an experimental environment to simulate corporate network traffic allowed for a more realistic and robust evaluation of the model. The results obtained in these scenarios, such as 93% precision in ransomware attacks and consistent performance in DoS attacks, evidence the model's applicability to real operational environments. This practical approach, combined with analyzing recognized datasets such as CICIDS2017 and UNSW-NB15, provides a solid framework to validate the system's effectiveness under diverse conditions.

However, it is important to recognize certain limitations of the study. Although the experimental environment enriched the analysis, evaluation in real networks with more significant heterogeneity could provide additional validation of the findings. Furthermore, the computational burden of training the model can be challenging on resource-constrained devices, such as those used in IoT networks. These constraints highlight key areas for improvement in future research.

This work contributes to cybersecurity by positioning GANs as a versatile and effective tool for detecting advanced threats. The model's ability to integrate malicious pattern detection and generation sets an important precedent, marking a clear direction toward more robust, adaptive, and efficient systems. This strengthens existing defenses and provides a framework for addressing emerging threats more quickly and accurately.

Future research will explore implementing the model in distributed environments, such as IoT networks, where scalability and computational efficiency will be crucial. Furthermore, optimizing the design of the generators and discriminators to handle multiple attacks simultaneously would be valuable, thus expanding the system's applicability. Another aspect is investigating explainability mechanisms for GANs in cybersecurity and generating trust and transparency in human operators' use of them.

7- Declarations

7-1- Author Contributions

Conceptualization, W.V.-Ch. and R.G.; methodology, W.V.-Ch.; software, W.V.-Ch.; validation, W.V.-Ch., R.G., and J.G.; formal analysis, W.V.-Ch.; investigation, W.V.-Ch.; resources, W.V.-Ch.; data curation, W.V.-Ch.; writing—original draft preparation, W.V.-Ch.; writing—review and editing, W.V.-Ch.; visualization, W.V.-Ch.; supervision, R.G.; project administration, R.G.; funding acquisition, R.G. All authors have read and agreed to the published version of the manuscript.

7-2- Data Availability Statement

The data presented in this study are available from the corresponding author.

7-3- Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

7-4- Institutional Review Board Statement

Not applicable.

7-5- Informed Consent Statement

Not applicable.

7-6- Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this manuscript. In addition, the ethical issues, including plagiarism, informed consent, misconduct, data fabrication and/or falsification, double publication and/or submission, and redundancies have been completely observed by the authors.

8- References

[1] Kamil, S., Siti Norul, H. S. A., Firdaus, A., & Usman, O. L. (2022). The Rise of Ransomware: A Review of Attacks, Detection Techniques, and Future Challenges. 2022 International Conference on Business Analytics for Technology and Security, ICBATS 2022. doi:10.1109/ICBATS54253.2022.9759000.

[2] Olawale, O. P., & Ebadinezhad, S. (2023). The Detection of Abnormal Behavior in Healthcare IoT Using IDS, CNN, and SVM. Lecture Notes on Data Engineering and Communications Technologies, 166, 375–394. doi:10.1007/978-981-99-0835-6_27.

[3] Pachika, S., Reddy, A. B., Pachika, B., & Karnam, A. (2024). Generative Adversarial Networks: Overview. Lecture Notes in Networks and Systems, 897, 319–328. doi:10.1007/978-981-99-9704-6_29.

[4] Shirazi, S. A. R., & Shaikh, M. (2024). A Novel Approach to Android Malware Intrusion Detection Using Zero-Shot Learning GANs. Sir Syed University Research Journal of Engineering & Technology, 13(2). doi:10.33317/ssurj.584.

[5] Li, Z., Wang, P., & Wang, Z. (2024). FlowGANAnomaly: Flow-Based Anomaly Network Intrusion Detection with Adversarial Learning. Chinese Journal of Electronics, 33(1), 58–71. doi:10.23919/cje.2022.00.173.

[6] Oyelakin, A. M. (2024). Overview and Exploratory Analyses of CICIDS2017 Intrusion Detection Dataset. Indonesian Journal of Data and Science, 4(3). doi:10.56705/ijodas.v4i3.80.

[7] More, S., Idrissi, M., Mahmoud, H., & Asyhari, A. T. (2024). Enhanced Intrusion Detection Systems Performance with UNSW-NB15 Data Analysis. Algorithms, 17(2). doi:10.3390/a17020064.

[8] Zhang, L., Bai, W., Xie, X., Chen, L., & Dong, P. (2024). TMANomaly: Time-Series Mutual Adversarial Networks for Industrial Anomaly Detection. IEEE Transactions on Industrial Informatics, 20(2), 2263–2271. doi:10.1109/TII.2023.3288226.

[9] Sun, H., Zhu, T., Zhang, Z., Jin, D., Xiong, P., & Zhou, W. (2023). Adversarial Attacks Against Deep Generative Models on Data: A Survey. IEEE Transactions on Knowledge and Data Engineering, 35(4), 3367–3388. doi:10.1109/TKDE.2021.3130903.

[10] Chougule, A., Agrawal, K., & Chamola, V. (2023). SCAN-GAN: Generative Adversarial Network Based Synthetic Data Generation Technique for Controller Area Network. IEEE Internet of Things Magazine, 6(3), 126–130. doi:10.1109/IOTM.001.2300013.

[11] Lim, W., Yong, K. S. C., Lau, B. T., & Tan, C. C. L. (2024). Future of generative adversarial networks (GAN) for anomaly detection in network security: A review. Computers and Security, 139. doi:10.1016/j.cose.2024.103733.

[12] Prabu, S., & Padmanabhan, J. (2024). Bi-channel hybrid GAN attention based anomaly detection system for multi-domain SDN environment. Journal of Intelligent and Fuzzy Systems, 46(1), 457–478. doi:10.3233/JIFS-233668.

[13] Agrawal, G., Kaur, A., & Myneni, S. (2024). A Review of Generative Models in Generating Synthetic Attack Data for Cybersecurity. Electronics (Switzerland), 13(2), 322. doi:10.3390/electronics13020322.

[14] Sayegh, H. R., Dong, W., & Al-madani, A. M. (2024). Enhanced Intrusion Detection with LSTM-Based Model, Feature Selection, and SMOTE for Imbalanced Data. Applied Sciences (Switzerland), 14(2), 479. doi:10.3390/app14020479.

[15] Almomani, O., Alsaaidah, A., Shareha, A. A. A., Alzaqebah, A., & Almomani, M. (2024). Performance Evaluation of Machine Learning Classifiers for Predicting Denial-of-Service Attack in Internet of Things. International Journal of Advanced Computer Science and Applications, 15(1), 263–271. doi:10.14569/IJACSA.2024.0150125.

[16] Vyšniūnas, T., Čeponis, D., Goranin, N., & Čenys, A. (2024). Risk-Based System-Call Sequence Grouping Method for Malware Intrusion Detection. Electronics (Switzerland), 13(1), 206. doi:10.3390/electronics13010206.

[17] Ibrahim Alsumaidaie, M. S., Ali Alheeti, K. M., & Alaloosy, A. K. (2024). An Assessment of Ensemble Voting Approaches, Random Forest, and Decision Tree Techniques in Detecting Distributed Denial of Service (DDoS) Attacks. Iraqi Journal for Electrical and Electronic Engineering, 20(1), 16–24. doi:10.37917/ijeee.20.1.2.

[18] Adiban, M., Siniscalchi, S. M., & Salvi, G. (2023). A step-by-step training method for multi generator GANs with application to anomaly detection and cybersecurity. Neurocomputing, 537, 296–308. doi:10.1016/j.neucom.2023.03.056.

[19] Wang, H., Miller, D. J., & Kesidis, G. (2023). Anomaly detection of adversarial examples using class-conditional generative adversarial networks. Computers and Security, 124. doi:10.1016/j.cose.2022.102956.

[20] Zerhouni, K., Gaba, G. S., Hedabou, M., Maksymyuk, T., Gurtov, A., & Amhoud, E. M. (2024). GAN-Based Evasion Attack in Filtered Multicarrier Waveforms Systems. IEEE Transactions on Machine Learning in Communications and Networking, 2, 210–220. doi:10.1109/tmlcn.2024.3361834.

[21] Choudhary, S., & Kesswani, N. (2020). Analysis of KDD-Cup'99, NSL-KDD and UNSW-NB15 Datasets using Deep Learning in IoT. Procedia Computer Science, 167, 1561–1573. doi:10.1016/j.procs.2020.03.367.

[22] Kumar, P., Bagga, H., Netam, B. S., & Uduthalapally, V. (2022). SAD-IoT: Security Analysis of DDoS Attacks in IoT Networks. Wireless Personal Communications, 122(1), 87–108. doi:10.1007/s11277-021-08890-6.

[23] Moustafa, N. (2021). The UNSW-NB15 Dataset | UNSW Research. The University of New South Wales, Sydney, Australia.

[24] Wireshark. (2019). Cybersecurity Blue Team Toolkit, 83–96, John Wiley & Sons, Hoboken, United States. doi:10.1002/9781119552963.ch7.

[25] Iqbal, R., Hussain, R., Arif, S., Ansari, N. M., & Shaikh, T. A. (2023). Data Analysis of Network Parameters for Secure Implementations of SDN-Based Firewall. Computers, Materials and Continua, 77, 1575–1598. doi:10.32604/cmc.2023.042432.

[26] Aziz, O., Klenk, J., Schwickert, L., Chiari, L., Becker, C., Park, E. J., Mori, G., & Robinovitch, S. N. (2017). Validation of accuracy of SVM-based fall detection system using real-world fall and non-fall datasets. PLoS ONE, 12(7), 0180318. doi:10.1371/journal.pone.0180318.

[27] Long, Z., & Jinsong, W. (2022). A hybrid method of entropy and SSAE-SVM based DDoS detection and mitigation mechanism in SDN. Computers and Security, 115. doi:10.1016/j.cose.2022.102604.

[28] de Carvalho, A. M. X., de Souza, M. R., Marques, T. B., de Souza, D. L., & de Souza, E. F. M. (2023). Familywise type I error of ANOVA and ANOVA on ranks in factorial experiments. Ciencia Rural, 53(7), e20220146. doi:10.1590/0103-8478cr20220146.

[29] Adnan, M., Imam, M. O., Javed, M. F., & Murtza, I. (2024). Improving spam email classification accuracy using ensemble techniques: a stacking approach. International Journal of Information Security, 23(1), 505–517. doi:10.1007/s10207-023-00756-1.

[30] Altulaihan, E., Almaiah, M. A., & Aljughaiman, A. (2024). Anomaly Detection IDS for Detecting DoS Attacks in IoT Networks Based on Machine Learning Algorithms. Sensors, 24(2), 713. doi:10.3390/s24020713.

[31] Zeeshan, M., Riaz, Q., Bilal, M. A., Shahzad, M. K., Jabeen, H., Haider, S. A., & Rahim, A. (2022). Protocol-Based Deep Intrusion Detection for DoS and DDoS Attacks Using UNSW-NB15 and Bot-IoT Data-Sets. IEEE Access, 10, 2269–2283. doi:10.1109/ACCESS.2021.3137201.

[32] Cekić, M. (2024). Anomaly Detection in Medical Time Series with Generative Adversarial Networks: A Selective Review. Anomaly Detection - Recent Advances, AI and ML Perspectives and Applications, IntechOpen, London, United Kingdom. doi:10.5772/intechopen.112582.