A Conditional Tabular Generative Adversarial Network (CTGAN)-based approach to safeguarding artificially created smart IoT settings

M.Raghu[1], Dr. S. Ganesh Kumar[2]
[1]Assistant Professor, Department of Electrical & Electronics Engineering,
Park college of Engineering and Technology-Coimbatore, Tamilnadu, India.
[2]Department of Data Science And Business Systems, Faculty of Engineering & Technology,
SRM Institute of Science and Technology, Kattankulathur, Tamilnadu, India.

Article Info

Article history:

Received Mar 29, 2024
Revised Apr 16 2024
Accepted May 27, 2024

Keywords:

Machine learning
Cyber security
Deep learning
Adversarial learning
Phishing

ABSTRACT

In the modern world, cyber security is essential since social media and mobile phones are used by all people and might lead to security issues. Machine learning algorithms, and more specifically deep learning algorithms, have seen widespread application in recent years in a variety of sectors, including cyber security. Initially, the stages at which adversarial attack tactics occur, as well as the goal and capabilities of the attacker, are used to characterize them. Next, we classify the ways that adversarial attack and defense techniques are used in the field of cyber security. Finally, we highlight certain aspects of recent research and analyze how recent advancements in other adversarial learning domains may impact future research directions in cyber security. This method depends heavily on using a Conditional Tabular Generative Adversarial Network (CTGAN), a powerful tool that learns from real data patterns to generate realistic synthetic network traffic. Implementing this generated information in a software-defined networking (SDN)-based simulated network environment is a crucial step in our strategy to evaluate and improve the traffic patterns. The resulting dataset's accuracy, using a decision tree, was 0.97, and its less complicated structure was obtained with training and test periods of 0.07 and 0.005 seconds, respectively. The findings demonstrate that synthetic data are appropriate for Internet of Things (IoT) contexts and smart city applications since they are simpler and correctly represent real data.

Corresponding Author:

Name of Corresponding Author,
Department of Electrical and Computer Engineering,
National Chung Cheng University.
Email: lsntl@ccu.edu.tw

1. INTRODUCTION

The interdependence of systems, companies, and society raises cyber security issues, which traditional models cannot adequately handle. The importance of cyber-security for both public and commercial businesses has increased due to the global growth in cyber-attacks and their substantial financial consequences. The effectiveness of traditional cyber security techniques, including rule-based firewalls and signature-based detection, in countering increasingly complex and dynamic cyber threats is frequently constrained [1]. It becomes computationally demanding to even manage the interactions of software installations inside a medium-sized business, taking into account the vulnerabilities linked to them in the software supply chain. The intricate nature of organizational cyber security is further highlighted by the addition of organizational security protocols, the ongoing adaptation of cyber attacker tactics, and the growing reliance on cloud services by third parties.

Gathering and storing network data for ML algorithm training is difficult in the IoT environment. One popular IoT protocol for connecting devices and enabling communication between them is message queuing telemetry transport (MQTT) [2]. To prevent network exposure and vulnerabilities to assaults by cybercriminals looking to steal sensitive data, MQTT broker security flaws must be fixed. Additionally, the MQTT protocol is simple to deploy in embedded devices due to its lightweight nature.

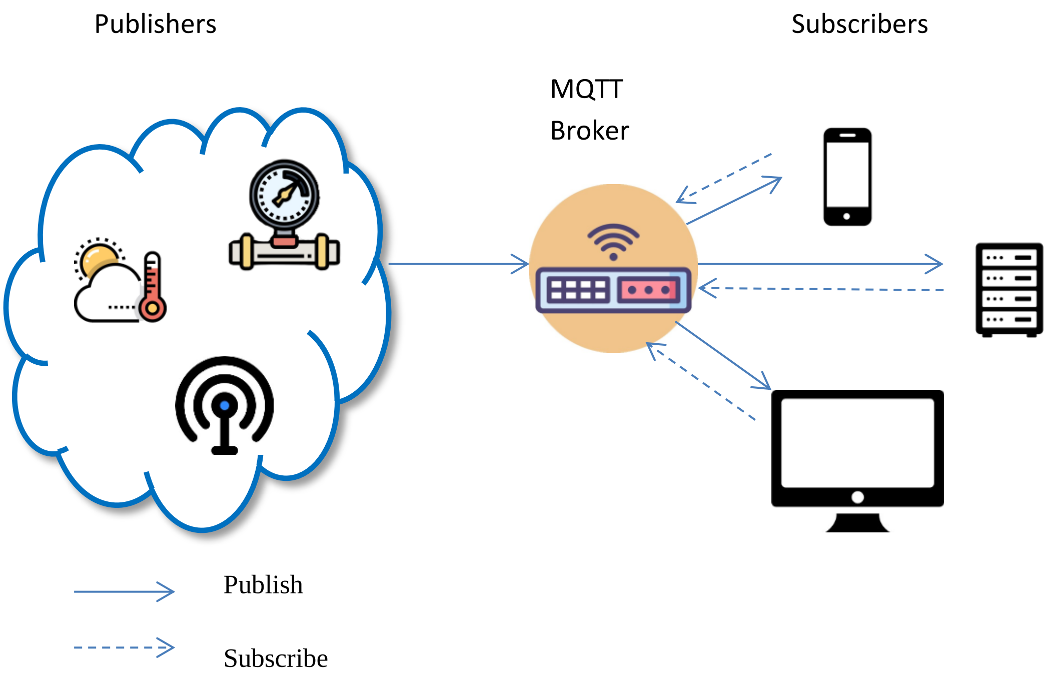

Figure 1. Architecture of MQTT

[3]The link between the publisher and the subscriber is explained by the MQTT architecture. MQTT brokers are used to transmit the messages from the publisher, which could be sensors, meters, etc. which enable the correct message to be sent to the right client. The subscriber's message is referred to as subscribe, and it is sent to the MQTT broker. These subscribers could be servers, PCs, mobile devices, etc.

Cyber security professionals have created training regimens suited to different jobs based on their responsibilities in attack and defense in order to solve this difficulty, and competitive mock training systems have been devised. [4] The use of generative adversarial networks (GANs) for deep learning-based data synthesis that replicates network traffic flows is becoming more popular in this context, even though technologies like schedulers and behavior modeling with Markov models can be used for automatic and realistic network traffic generation [5]. The conditional tabular generative adversarial network (CTGAN), an essential component of the proposed system, generates synthetic tabular data with statistical properties equivalent to those of actual data. Features like packet size, transmission time, protocol type, and source and destination IP (Internet Protocol) addresses are examples of statistical characteristics in a network environment.

[6]CTGAN analyzes the statistical properties and structure of actual datasets to produce synthetic tabular data. The time and expense of gathering and storing the dataset on a big storage device are decreased when a synthetic dataset is used. This would assist in resolving the difficulties associated with gathering network data in an Internet of Things setting. The time and expense of gathering and storing the dataset on a big storage device are decreased when a synthetic dataset is used. This would assist in resolving the difficulties associated with gathering network data in an IoT setting. Some of the most often used techniques for analyzing synthetic network traffic data. Three criteria (bandwidth, accuracy, and precision) were used in the research to evaluate the rate control capabilities of the investigated packet generators in order to control the generated traffic to conform to particular specifications [7]. A brief note on this:

- Bandwidth: The speed of packet transmission expressed in packets per second, or the maximum transmission capacity attained throughout the generating process.
- Accuracy: Measures how close the observed average rate is to the set rate and refers to systematic errors, which stand for statistical bias.
- Precision: Defines the divergence of individual inter-packet gaps from the predetermined value and refers to random errors, which quantify statistical unpredictability [8].

Features such as packet size, transmission time, protocol type, and source and destination IP (Internet Protocol) addresses are examples of statistical characteristics in a network environment. For evaluating the amount of traffic coming from hosts, a network traffic matrix can be created and examined [9]. Hence providing crucial information about changes and improvements needed to achieve realistic cyber security exercise scenarios. Therefore, the efficiency of the suggested strategy in optimizing network performance, creating cyber security drill scenarios and confirming the efficacy of the traffic flowing through the training and research platforms for cyber security [10].

2. LITERATURE REVIEW

Kim et al. [11] proposed Network Traffic Synthesis and Simulation Framework for Cyber security Exercise Systems. CTGAN, a complex tool that creates realistic synthetic network traffic by learning from actual data patterns, is the core component of the suggested methodology. This technology makes it possible to handle sensitive data and technical components with high constancy. Making certain that the artificial data has statistical traits akin to those found in actual network surroundings. Consequently, it is crucial to use models like CTGAN to enhance the quality of synthetic data and raise its statistical resemblance to real data.

Adiputra et al. [12] designed CTGAN-ENN: a tabular GAN-based hybrid sampling method for imbalanced and overlapped data in customer churn prediction. This study compared CTGAN-ENN with popular over-sampling and hybrid sampling techniques. For stakeholders who wish to incorporate customer attrition projections into their business data, this paper has useful implications. Customer data is expanding quickly in everyday life, and the study's findings might shed light on big data domains. In order to create a customer churn prediction model, stakeholders can take into account their resources by selecting the best combination of CTGAN-ENN and machine learning algorithms.

Anande et al. [13] introduced generative adversarial networks for network traffic feature generation. In order to compare and assess the generated (synthetic) data with the real network traffic data for resemblance, distance between the generated and real distributions, and degree of difficulty in differentiating the two data streams, the study used the Synthetic Data Evaluation Framework. In this case, just 28% of the Vanilla GAN's features could be used in place of those in the actual data without being detected; however, this increased to 55% for CTGAN (69% when categorical features were converted) and over 85% for Copula GAN.

Apruzzese et al. [14] proposed the role of machine learning in cyber security. As data analytics methods advanced, detection systems started to use data-driven solutions like machine learning. In addition to requiring less manual labor, these systems occasionally even outperformed conventional handwritten detection schemes. Lastly, the study offers two case studies of effective industrial ML deployments that are now in use to combat cyber threats. This paper aims to drive significant advancements in machine learning within the cyber security field, paving the way for a broader application of ML solutions to protect current and future systems.

Vallabhaneni et al. [15] presented the detection of cyber security using bidirectional generative adversarial networks. Finding network threats is a key problem in cyber security detection, and several studies about IDS have been published. This paper's contribution is to provide the most recent dataset that is only linked to IoT-based attack behaviors, leaving out the behaviors that are typical of traditional models. This study also presents a novel generative DL-based intrusion detection system (IDS) model for efficiently countering cyber-attacks and improving classification performance.

Hintaw et al. [16] introduced MQTT vulnerability, attack vectors, and solutions in the Internet of Things. The MQTT protocol's principles and security features were explained. A detailed threat model of the MQTT protocol with attack vectors relevant to a traditional MQTT-based Internet of Things domain was described in this paper. Additionally, this study outlined current research problems and existing projects that will be addressed shortly. Additionally, as potential solutions to secure MQTT, the researchers described cutting-edge technologies like blockchain, artificial intelligence, and machine learning in conjunction with the Internet of Things.

3. PROPOSED METHODOLOGY

This study's primary goal is to provide cyber security using network traffic synthesis and simulation. Implementing traffic that simulates a real-world internet environment in the exercise communication network, including background traffic generation and user training, is crucial after cyber security exercise scenarios are ready. A conditional generator strategy is recommended since the previous generator method ignores categorical variable imbalance.

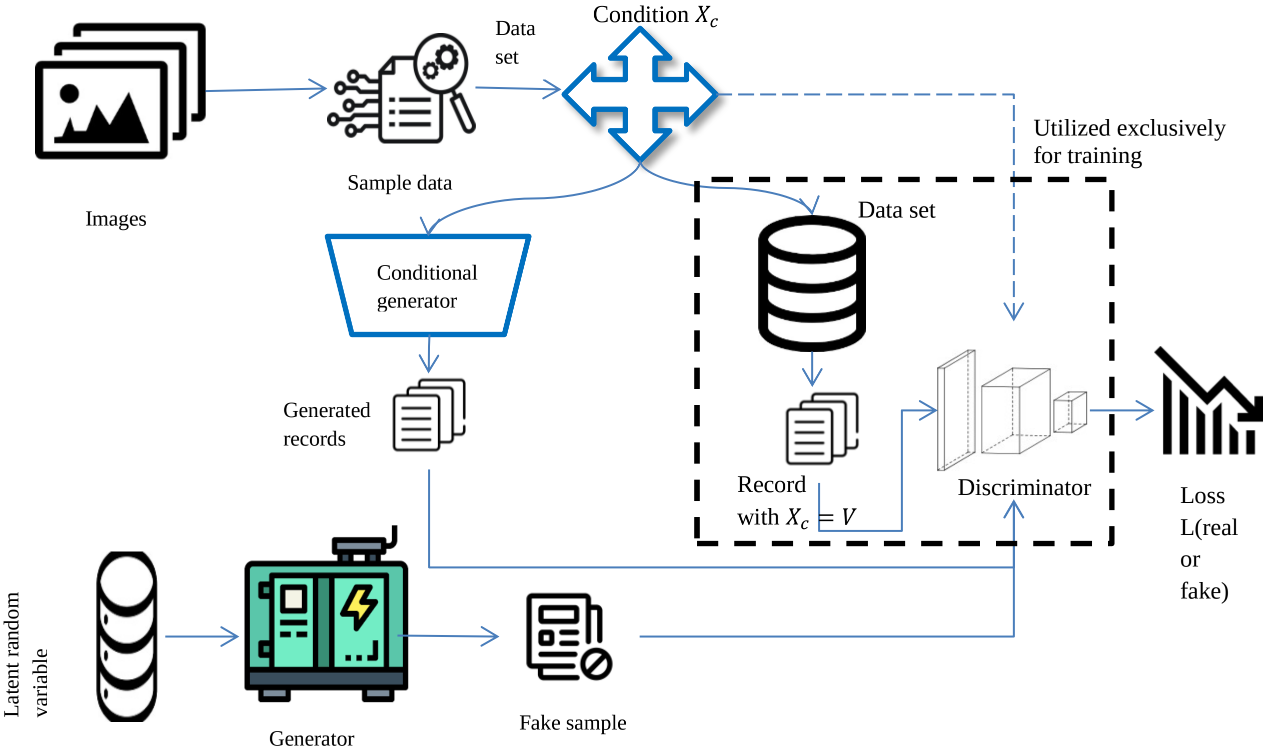

Figure 2. Architecture of CTGAN

Figure 2 is the architecture of CTGAN, which consists of multiple components that produce true and real output, as depicted in the above diagram. Images are among the components that allow the sample data to be driven; false data is created using generators from the latent random variable and sent to the discriminator, which provides an output to the decision maker that interprets the decision as a loss. To create a dataset that includes multiple attacks and stages and accurately reflects the real dataset, the CTGAN model was trained across 40–110 epochs, choosing the best test score based on the statistical verification measure. The dimensions of the discriminator and generator were (230,230), and their learning rates were fixed at 20,000 (Table 1).

Table 1. CTGAN parameter configuration

CTGAN Parameters	Values
Pace of generator learning	0.001
Generator's size	230
Discriminator's size	230
Epoch	40 to 110

Several popular machine learning classification techniques, such as decision trees, were utilized to assess the created dataset based on accuracy and training/testing timeframes. A default configuration setting for RF, decision tree, and other classifiers was used, taking into account their hyperparameters. Training and testing times in seconds, which show the complexity of the classification process; accuracy, a measure of the classifier's overall performance as a percentage of correct estimates; as well as the F1 score, that combines recall and precision, were the performance metrics employed. Each classifier was trained autonomously.

The performance metrics are described as follows: Attack instances that are correctly classified as attacks are called true positives (TP); normal instances that are correctly detected as normal are called true negatives (TN); normal instances that are mistakenly classified as attacks are called false positives (FP); and attack instances that are mistakenly classified as normal are called false negatives (FN). The performance metrics can be found using the following formulas:

$$F(score) = \frac{2TP}{2TP + FN + FP} \qquad (1)$$

This proposed system used an SDN (Software-Defined Networking) controller to regulate and send traffic throughout the network in order to compare and assess genuine packets with synthetic ones. Real-time monitoring and assessment of network traffic volumes were made possible by the SDN controller's communication with network switches to govern traffic. This system made it possible to create realistic cybersecurity training environments by offering an extensive technique for creating and distributing network training traffic. In order to solve the imbalance problem and balance malicious and authentic traffic, we used CTGAN to rebuild legitimate packets expressed as structured data.

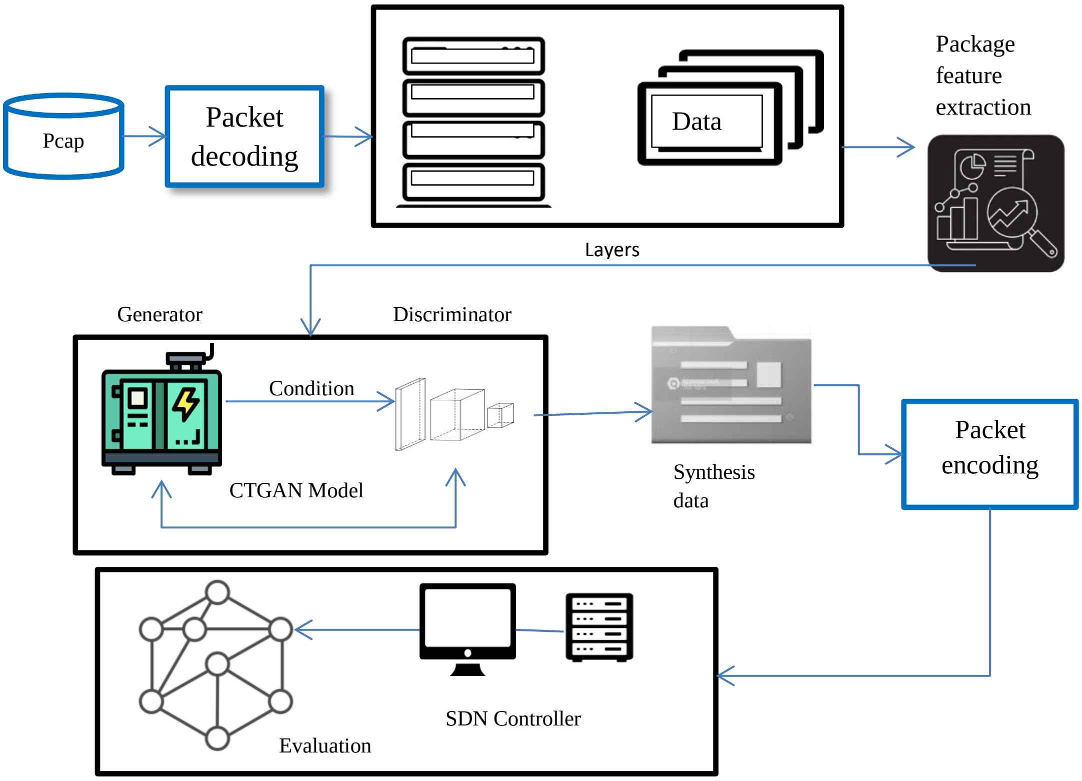

Figure 3. Network traffic generation

By the procedure shown in Figure 3, our system provided an extensive technique for generating and allocating network training traffic, allowing the development of realistic cybersecurity training environments. Understanding and extracting information from binary data packets transmitted over a network is known as decoding network packets. Captured packet data can be extracted from pcap files for packet decoding using tools such as Wireshark and TCPDUMP, which provide hexadecimal information showing the structure and content of data packets in network connections. CTGAN generates synthetic tabular data by mimicking the organizational structure and statistical characteristics of real datasets. By generating new instances while maintaining the statistical properties of the original real data, CTGAN can handle a variety of data types, including continuous, discrete, and categorical variables. The generator learns the conditional distribution of rows that match particular values of different columns under specified circumstances in order to reproduce a genuine data distribution. $\check{r} \sim p_g(row|\, d_{i^*} = k^*)$is the expression for $D_{r^*} = K^*$. Consequently, the conditional CTGAN can successfully reproduce the actual data distribution in the manner shown below.

$$P_{(row)} = \check{r} \sim p_g(row|\, d_{i^*} = k^*) \qquad (2)$$

The procedure for determining the overall distribution of the produced rows P(row) across all feasible circumstances k is explained in Equation (2). The conditional distribution of rows produced by generator G given condition K^* is represented by each phrase $p_g(row|\, d_{i^*} = k^*)$, and the probability in the actual data for condition K^*. is represented by $D_{r^*} = K^*$. CTGAN is capable of handling structured network traffic data and correctly reconstructing TCP traffic flow sequences. Several traffic logs are gathered and organized based on particular TCP streams.

Packed the tabular data produced by CTGAN into packets based on certain characteristics. First, the Ethernet layer includes data like MAC addresses and frame formats that enable packets to travel over the physical network to their recipients. Second, the IP layer contains the data needed for appropriate packet routing over the internet, such as source and destination IP addresses and packet length. Finally, the data segment contains the actual application data that is being provided. Here, the rate of change in traffic has been determined with the Present Traffic (PT) and Last Traffic (LT).

$$Rate\ of\ change\ in\ traffic(\Delta t) = \frac{LT - PT}{time} \qquad (3)$$

In Equation 3 we measure the time difference between two measurements and use the difference between present and before traffic states as the basis for our computations to identify the change in network traffic. In order to ascertain and modify the dynamic network performance, this approach enables real-time comprehension and response to changes in network conditions.

4. RESULT AND DISCUSSION

The information we utilized here covers every phase of a cyber-attack, from reconnaissance and scanning to exploitation. The collection is made up of 6 distinct files, the majority of which are packet captures from the surveillance and information-collecting stages. We created synthetic traffic using the final file (maccdc2012_00016.pcap), which has 3,816,907 traffic records, rather than using the complete dataset. To synthesize data, we chose some top hosts in the network from this dataset.

Instead of using all the available hosts, it is probably more convenient and appropriate to use a subset of traffic flow scenarios for synthetic data evaluation. This allows the synthetic data to be incorporated into the original traffic flows in cyber security exercise scenarios.

In accordance with communication scenarios in the training system, the system was set up to examine and mimic traffic patterns in diverse network contexts.

Table 2. Host, number, and size of packets

Host	No. of packets	Size
192.168.202.108	43435	672352645
192.168.23.656	6564	7543234790
192.168.27.234	4561	24235688
192.168.21.202	254	38551

The data in the above table is provided based on the IP address and the number of packets sent and received in a particular size.

We utilized CTGAN to synthesize data based on the original pcap file, taking into account the environment and dataset configuration.

The generator featured a learning rate of 0.001 and three layers with dimensions of 230, 230, and 1. With three layers with dimensions of 120, 120, and 1, the discriminator effectively distinguished between synthetic and real data despite having a lower capacity than the generator. By minimizing superfluous training, the smaller discriminator improved computing efficiency and avoided overfitting. The graph displays the CTGAN-generated synthetic traffic data for comparison. The artificial traffic data exhibits variations that are comparable to the real traffic data, suggesting that they reflect actual network activity.

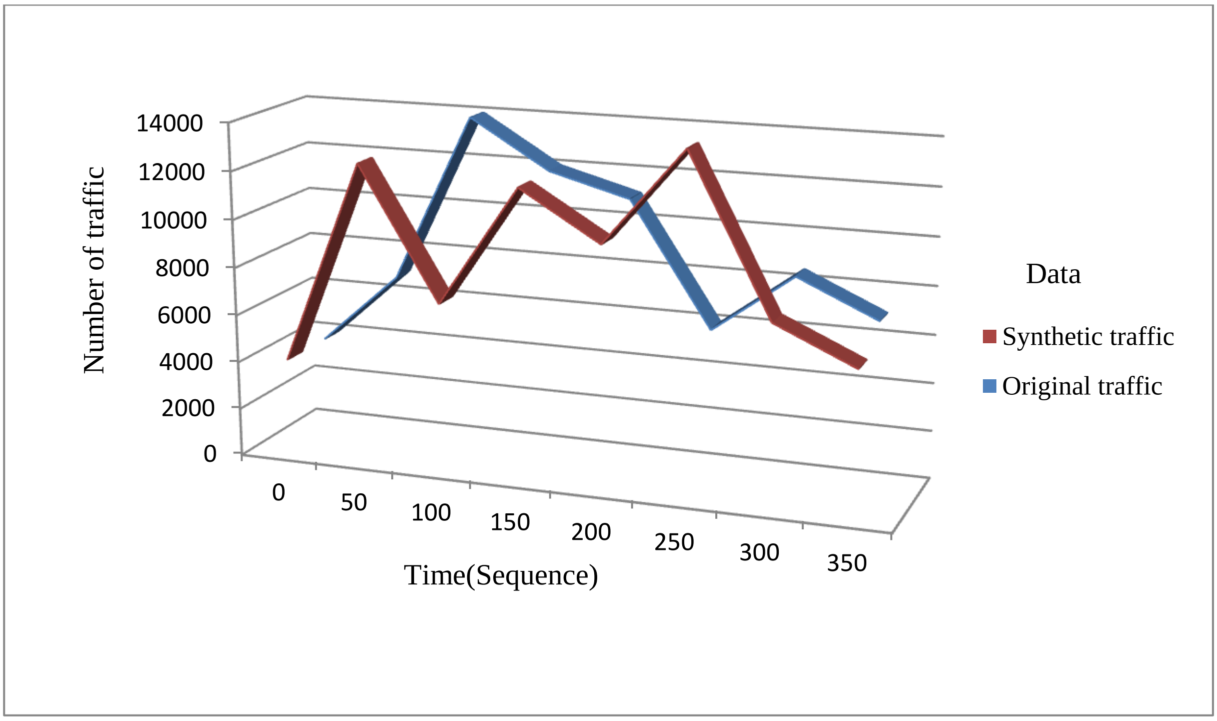

Figure 4. Traffic Accuracy

The real and synthetic traffic curves in Figure 4 exhibit maxima of high traffic numbers at particular times. These highs happen when there is a lot of data being transmitted across the network. Synthetic traffic accurately depicts an actual environment, as seen by the similarities between the varying patterns of the two types of traffic data.

To determine the percentage difference, we first computed the difference in the total traffic volume between the synthetic and actual data, and then we divided that difference by the total traffic volume of the real data. This computation is provided by:

$$P = \sqrt{\frac{D_1 - D_2}{D_1}} \quad (4)$$

The entire volume of synthetic and real traffic is denoted by D_1 and D_2, respectively. An easy explanation of the changes over time is made possible by the calculation, which displays the percentage difference in total traffic volume between real and synthetic data.

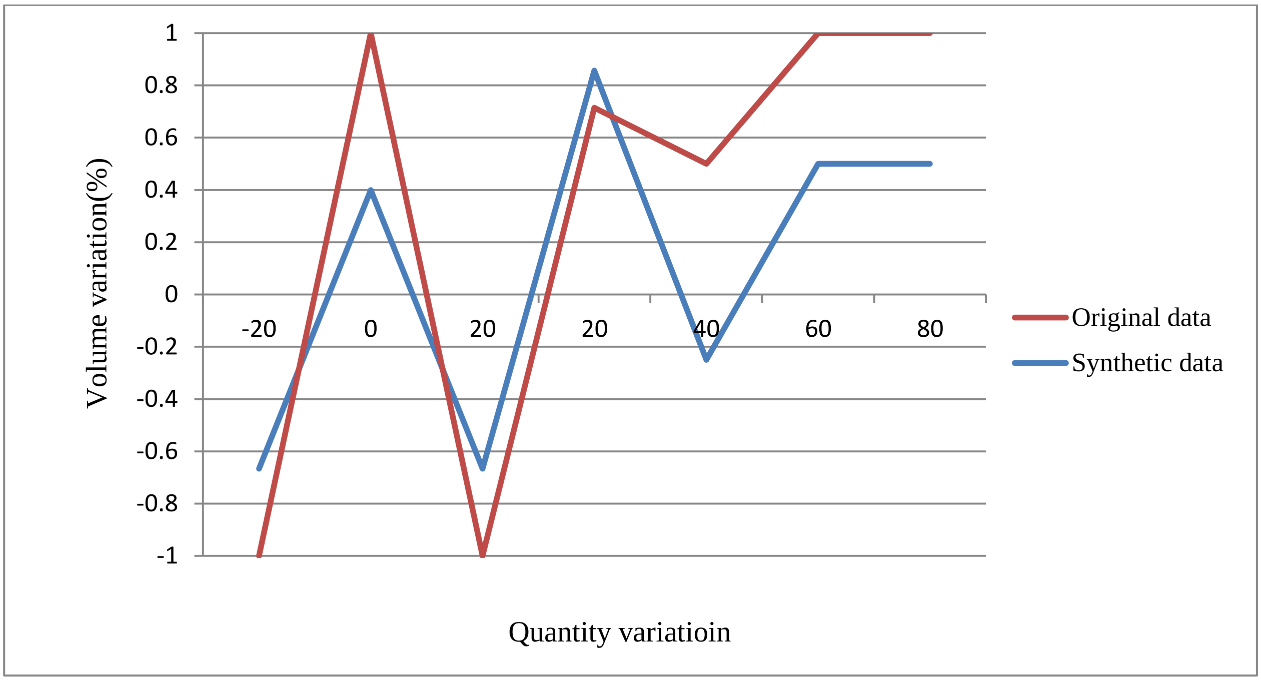

Figure 5. Traffic rates between synthetic and real data

In Figure 5 Analysis was done on the variations in the volume and quantity of real and synthetic packets. The total difference between the real and synthetic traffic data is quantified by calculating the average of the percentage differences, which shows the average relative difference between the two sets of data. The remaining 70.72% of synthetic data is thought to exhibit patterns that are comparable to those of real data since an average percentage difference of 29.28% suggests a significant difference between real and synthetic data.

Table 3. The quantity and size of packets

Host	No. of real packets	No of synthetic packets	Actual data size	Synthetic data size	Loss(%)
192.168.202.2	32546	324	3,323,213	2,324,346	8.64%
192.168.02.8	2435	23446	7,654,368	3,246,658	20.32%
192.168.128	678658	53412	3,571,287	241,225	12%
192.168.27.536	243	21312	4,643	2,143	14.5%

Table 3 shows that host 192.168.202.2 has the highest byte loss rate at 36.77%. This indicates that an important part of data is not transmitted, which has a major impact on the volume shown in Figure 4. This issue seems to be caused by improperly supplied encrypted payloads during network layer encoding. This could be brought on by problems encoding encrypted payloads or by other network problems. Despite these variations, synthetic traffic retains the general volume and patterns of real traffic, indicating that other faults occur consistently.

5. CONCLUSION

The study's findings establish that the proposed approach can potentially be used to create a new synthetic dataset with a higher number of attacks without the need for months of recording or a lot of storage space to gather attack datasets. Additionally, the feature-selected dataset may be used with ML-based IDS in a lightweight IoT context without compromising the accuracy of inter-device communication or real-time updates. high traffic volumes at particular points in the real and synthetic traffic curves, suggesting that the synthetic traffic more accurately depicts the real network environment than it did previously. Providing cyber security trainees realistic circumstances and enhancing their ability to identify and respond to realistic dangerous packets need lowering the mistake rate. As network traffic standards increase, future research will focus on evaluating synthetic packet data and improving monitoring tools. The successful development of iterative technological frameworks for cybersecurity training systems is ensured by the usage and verification of these improvements in real-world simulation training systems for a wider range of scenarios.

REFERENCES

[1] Das, S. (2022). FGAN: Federated generative adversarial networks for anomaly detection in network traffic. arXiv preprint arXiv:2203.11106.Alabdulwahab, S., Kim, Y. T., Seo, A., & Son, Y. (2023). Generating Synthetic Dataset for ML-Based IDS Using CTGAN and Feature Selection to Protect Smart IoT Environments. *Applied Sciences*, *13*(19), 10951.

[2] Kholgh, D. K., & Kostakos, P. (2023). PAC-GPT: A novel approach to generating synthetic network traffic with GPT-3. *IEEE Access*.

[3] Oh, S. H., Jeong, M. K., Kim, H. C., & Park, J. (2023). Applying Reinforcement Learning for Enhanced Cybersecurity against Adversarial Simulation. *Sensors*, *23*(6), 3000.

[4] Mendikowski, M., & Hartwig, M. (2022). Creating customers that never existed: Synthesis of e-commerce data using CTGAN. In *18th International Conference on Machine Learning and Data Mining (MLDM-22). New York, US: IBAI Publishing* (pp. 91-105).

[5] Lee, J. S., & Lee, O. (2021). Ctgan vs tgan? which one is more suitable for generating synthetic eeg data. *J. Theor. Appl. Inf. Technol*, *99*(10), 2359-2372.

[6] Oh, S. H., Kim, J., & Park, J. (2024). Dynamic Cyberattack Simulation: Integrating Improved Deep Reinforcement Learning with the MITRE-ATT&CK Framework. *Electronics*, *13*(14), 2831.

[7] Hang, C. N., Yu, P. D., Morabito, R., & Tan, C. W. (2024). Large Language Models Meet Next-Generation Networking Technologies: A Review. *Future Internet*, *16*(10), 365.

[8] Zacharis, A., Katos, V., & Patsakis, C. (2024). Integrating AI-driven threat intelligence and forecasting in the cyber security exercise content generation lifecycle. *International Journal of Information Security*, 1-20.

[9] Garcia-Ortiz, A., Amin, S. M., & Wootton, J. R. (1995). Intelligent transportation systems—Enabling technologies. *Mathematical and Computer Modelling*, *22*(4-7), 11-81.

[10] Kim, D. W., Sin, G. Y., Kim, K., Kang, J., Im, S. Y., & Han, M. M. (2024). Network Traffic Synthesis and Simulation Framework for Cybersecurity Exercise Systems. *Computers, Materials & Continua*, *80*(3).

[11] Adiputra, I. N. M., & Wanchai, P. (2024). CTGAN-ENN: a tabular GAN-based hybrid sampling method for imbalanced and overlapped data in customer churn prediction. *Journal of Big Data*, *11*(1), 121.

[12] Anande, T. J., Al-Saadi, S., & Leeson, M. S. (2023). Generative adversarial networks for network traffic feature generation. *International Journal of Computers and Applications*, *45*(4), 297-305.

[13] Apruzzese, G., Laskov, P., Montes de Oca, E., Mallouli, W., Brdalo Rapa, L., Grammatopoulos, A. V., & Di Franco, F. (2023). The role of machine learning in cybersecurity. *Digital Threats: Research and Practice*, *4*(1), 1-38.

[14] Vallabhaneni, R., Vaddadi, S. A., Pillai, S. E. V. S., Addula, S. R., & Ananthan, B. (2024). Detection of cyberattacks using bidirectional generative adversarial network. *Indonesian Journal of Electrical Engineering and Computer Science*, *35*(3), 1653-1660.

[15] Hintaw, A. J., Manickam, S., Aboalmaaly, M. F., & Karuppayah, S. (2023). MQTT vulnerabilities, attack vectors and solutions in the internet of things (IoT). *IETE Journal of Research*, *69*(6), 3368-3397.