Enhancing Cloud Security Using Generative AI for Intrusion Detection

Vineet Kumar Srivastava[1*], Vishnu Ravi[2], Maninder Pal Singh[3], Nuzhat Noor Islam Prova[4]

[1*]Sr. Software Engineer, Peoria, Arizona, 85382, USA.
[2]Lead Software Engineer, Bayonne, New Jersey, 07002, USA.
[3]Lead Software Engineer, Princeton, New Jersey, 08540, USA.
[4]Sr. Data Scientist, Queens, New York, 11432, USA.

*Corresponding author(s). E-mail(s): icyvineet@gmail.com;
Contributing authors: vishnu3186@gmail.com; mpsgotra85@gmail.com; nuzhatnsu@gmail.com;

Abstract

Cloud computing has revolutionized data processing, while it is still quite fairly vulnerable to advanced cyberattacks, including DDoS attacks and zero-day flaws. Regarding changing assault methods, data imbalance, and real-time threat detection, traditional intrusion detection systems (IDSs) struggle. This paper presents a Generative AI-enhanced Intrusion Detection System (GAI-IDS) to improve threat detection accuracy and resilience by combining Transformer-based anomaly detection with Conditional Generative Adversarial Networks (CGANs). The model balances insufficiently represented threat classes, provides reasonable synthetic attack samples, and uses a multi-head self-attention technique for real-time anomaly detection. With a 96.5% detection accuracy, experimental results on the ToN-IoT dataset much exceed conventional IDS. Moreover, adversarial training builds up the system to resist evasive cyberattacks, providing scalable and flexible cloud security.

Keywords: Cloud Security, Intrusion detection, Generative AI, Cyber attacks, ToN-IoT, Denial-of-service (DoS)

R. Nagariya et al. (eds.), *Proceedings of the International Conference on Sustainable Business Practices and Innovative Models (ICSBPIM-2025)*, Advances in Economics, Business and Management Research 349,
https://doi.org/10.2991/978-94-6463-872-1_50

1 Introduction

Cloud computing has transformed data access, storage, and processing for companies. Rising at a compound annual growth rate (CAGR) of 18.0%, from 2020 to 2025, the worldwide cloud computing market is predicted to reach $623.3 billion by 2025 [1]. Including zero-day exploits, data breaches, and distributed denial-of-service (DDoS) events, advanced threats challenge traditional IDSs to keep pace. These difficulties need advanced security systems able to instantly recognize and reduce risks [2].

Rising as an effective tool in cybersecurity, Generative AI provides proficient threat detection features by learning attack patterns, creating synthetic threat circumstances, and improving anomaly detection models [3]. By organizing important issues including data imbalance, adversarial evasion, and real-time detection in high-traffic cloud environments, techniques including Generative Adversarial Networks (GANs), Variational Autoencoders (VAEs), and Transformer-based models have shown the ability to considerably increase the accuracy and efficiency of IDS [4]. In contrast to conventional IDS, which depend on predefined patterns or methods, AI-driven security systems can adapt to changing attack patterns, so they are quite successful against unidentified attacks [5].

We present a GAI-IDS combining Transformer-based anomaly detection in cloud environments with Conditional GANs (CGANs) for generating synthetic attack data. By means of realistic attack sample generation, synthetic augmenting of underrepresented attack classes, and real-time scalable detection enabled by a multi-head self-attention mechanism, our model is intended to overcome the constraints of conventional IDS by enhancing threat detection accuracy and so mitigating data imbalance issues. Our method further increases security resilience by including adversarial training methods, therefore avoiding the deception strategies utilized by powerful hackers. The key contributions of this research are:

1. We offered a GAI-IDS using Transformers and Conditional GANs (CGANs) to identify advanced cyber threats, thereby improving cloud security.
2. We used CGANs to create feasible attack samples, therefore reducing data imbalance and greatly enhancing the zero-day attack and underprivileged background's threat detection abilities.
3. We provided scalability, efficiency, and adaptability in dynamic cloud environments by implementing a Transformer-based IDS with multi-head self-attention for real-time anomaly detection.

The remainder of this paper is structured as follows: section 2 presents a broad review of existing related work in Generative AI for Intrusion Detection. In section 3 we describe the methodology proposed: data preprocessing, model architecture, and optimization techniques. In section 4, we present and discuss our experimental results, benchmarking against other state-of-the-art methods. Finally, section 5 concludes this paper, with future research directions outlined.

2 Literature Review

Recent studies have looked at the impact of Generative AI in increasing Security, intrusion detection, and data privacy across numerous areas. They have developed AI-driven frameworks merging GANs, federated learning, and VAEs to better anomaly detection and threat mitigation. They reported getting greater accuracy, but obstacles such as processing overhead, adversarial threats, and real-time adaptation remain important limits.

Khanna et al. [1] investigated how automated reactions, threat intelligence, and anomaly detection made possible by generative artificial intelligence might improve cloud security. Reviewing artificial intelligence methods including GANs and VAEs, shows how they can be used to identify cyber threats, create synthetic security datasets, and automatically implement security policies High accuracy case studies including 95.7% anomaly detection, an F1-score of 0.93, and a 24% improvement in threat detection are analyzed. Park et al. [6] demonstrated an AI-based network intrusion detection system (NIDS) using GANs to solve the data imbalance issue in intrusion detection. To create synthetic attack traffic and enhance classification performance, the authors developed a framework combining a Wasserstein distance-based GAN with autoencoder-driven deep learning models. Their approach was assessed using real-world network data, IoT-23, NSL-KDD, and UNSW-NB15 among other datasets. Outperformance of past deep learning-based NIDS, the suggested system attained up to 93.2% accuracy on NSL-KDD and 87% on UNSW-NB15. Nadella et al. [7] proposed a Generative AI-based cybersecurity architecture combining GANs, VAEs, and anomaly detection models, to improve enterprise data privacy management. To guarantee data privacy using differential privacy, encryption, and data masking, they created a system employing Random Forest, SVMs, and LSTM networks for threat detection. Balancing accuracy, memory, and computing efficiency, their trials revealed great accuracy in the financial (94%), healthcare (96%), and smart city (91%), sectors. Dunmore et al. [8] used GANs in cybersecurity, especially for IDSs are examined. They looked at how GANs are applied for developing polymorphic malware samples, data augmentation for unusual attack classes, and adversarial example creation. Along with datasets including NSL-KDD, CIC-IDS, and IoT-23, the paper examines several GAN designs including WGANs, BiGANs, and DCGANs. Depending on the dataset and model variation, reported IDS accuracy increases span from 88.23% to 99.73%. Furthermore, Lopez et al. [9] offered a thorough overview of GAN AI applications in IoT security together with a discussion of how models powered by AI improve cybersecurity in linked devices. They investigated IoT networks and the application of GANs, VAEs, and Transformer-based models for intrusion detection, anomaly detection, encryption, and access control. They assess many AI-driven security models and datasets applied for model development. Although particular accuracy measures differ depending on the application, several mentioned studies show increases in intrusion detection rates between 88% and 99%. Also, Kumaran et al. [10] identified software piracy and intrusion attempts and presented a GAN-IF-based cybersecurity model combining GANs with Isolation Forests (IF). While a discriminator separates between original and pirated programs, the model trains a GAN generator to replicate legal software behavior. The Isolation Forest picks for anomalies connected to

illegal software use. The outperformance of other methods like One-Class SVM, Random Forest, and RNN is shown by experimental data displaying 98.5% accuracy, 92% recall, and 95% precision. Likewise, Andreoni et al. [11] investigated to enhance security, trustworthiness, and resilience in autonomous systems including robotic systems, self-driving cars, UAVs, and GenAI. It goes over how GANs, VAEs, Transformers, and LLMs enhance intrusion detection, predictive maintenance, adaptive traffic management, and threat response. The paper notes claimed accuracy gains in several cybersecurity applications ranging from 88% to 99%. Vu et al. [12] offered a thorough assessment of GAI in mobile and wireless networking covering domains like network administration, wireless security, and semantic communication. They looked at how GAI models—including Transformers, VAEs, and GANs—improve network resource allocation, intrusion detection, traffic management, and threat-mitigating ability. From 88% to 99% in cybersecurity applications like anomaly detection and wireless intrusion detection, several research mentioned in the paper show accuracy gains. Mohialden et al. [13] suggested an AI-driven cybersecurity framework for shielding healthcare data from data poisoning assaults, using federated learning, homomorphic encryption, and autoencoder-based anomaly detection. The strategy guarantees distributed model training, privacy-preserving calculations, and enhanced anomaly detection in medical data. Strong security performance with an accuracy of 97.8%, recall of 94.5%, and an F1-score of 96.2%—was shown by tests utilizing simulated healthcare datasets.

The lack of enhancing cloud security in previous studies thair are some several limitations:

1. On benchmark sets, models show great accuracy; but, they suffer from zero-day attacks changing threats, and real-world implementation.
2. GANs, VAEs, and federated learning require large resources, so IoT applications and real-time intrusion detection are not feasible.
3. AI-driven security models remain vulnerable to adversary manipulation, thereby probably avoiding anomaly and IDSs.
4. As black boxes, DL-based cybersecurity models prevent trust, regulatory acceptance, and debugging in security-critical applications.

3 Methodology

This study presents an IDS using a Transformer for accurately classifying network traffic and a CGAN for generating synthetic attack samples. This section details how the methodology has followed a systematic approach in data preprocessing, synthetic data augmentation, intrusion detection modeling, and anomaly detection to augment the security of cloud-based infrastructures. As shown in Figure 1, the workflow integrates these components in order to enhance the intrusion detection accuracy and zero-day attack identification in the cloud environment.

3.1 Dataset Description

The ToN-IoT dataset, a high-dimensional intrusion detection data set, is utilized in this study, and it is curated specifically to evaluate the cybersecurity models in cloud

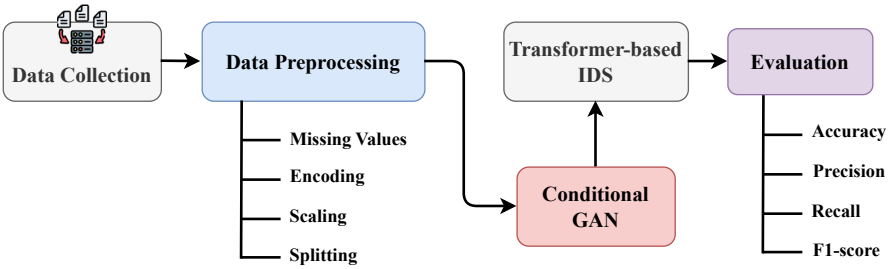

Fig. 1: Overview of the Proposed Hybrid Intrusion Detection Framework

and IoT-based networks brought from Kaggle. It has been constructed using 45 different network attribute features with 461,043 network traffic samples, which include various network attributes, such as packet size, source and destination IP addresses, protocol types, and traffic flow characteristics. The dataset mirrors real cyber threats in the form of both normal traffic and ten different attack classes such as scanning, injection, DoS, DDoS, password attack, ransomware, man in the middle (MITM), cross-site scripting (XSS), and backdoor. For the purpose of training robust and scalable intrusion detection models to secure cloud-based infrastructures, it is highly suitable due to the diversity of attack classes.

3.2 Dataset Preprocessing

Different preprocessing steps were used on the dataset to prepare it for training such that features were well represented to the model so that both feature representation and the model's performance can be optimized.

1. **Handling Missing Values:** The numerical features were completed with missing entries using mean imputation to make the data complete as follows:

$$x_{i,j} = \frac{1}{m} \sum_{i=1}^{m} x_{i,j}, \quad \text{if } x_{i,j} \text{ is missing} \tag{1}$$

2. **Encoding Categorical Variables:** One-hot encoding converts feature values on protocol types and attack labels to numerical values for the purposes of making categorical features binary vectors.
3. **Feature Scaling:** Min-Max normalization was applied for standardizing feature ranges as follows:

$$x'_j = \frac{x_j - \min(x_j)}{\max(x_j) - \min(x_j)} \tag{2}$$

4. **Dataset Splitting:** For training and testing of the dataset, the scikit-learn's train_test_split method was used and the dataset was divided into 80% training and 20% testing set.

3.3 Synthetic Data Generation Using GANs

The input of the generator is a random noise vector z sampled from a normal distribution, class labels y, corresponding to the attack type. Batch normalization and LeakyReLU are then added as initialization to keep training from getting stuck in bad solutions and prevent mode collapse, and the input goes through several layers that fully connect the input. The statistics of real traffic are mimicked to some degree by the generator's output feature vector. The generator's objective function is defined as:

$$L_{\mathrm{G}} = -\mathbb{E}[\log D(G(z, y), y)] \tag{3}$$

where $G(z, y)$ is the generated attack sample and $D(G(z, y), y)$ is the discriminator's prediction of whether the sample is real.

On the other hand, the discriminator is dealing with the same problem as a binary classifier, separating real samples from generated ones. It also gets the real network traffic samples as well as the synthetic samples from the CGAN, along with their labels. The complex decision boundaries are learned through these multiple fully connected layers with LeakyReLU activation. The last layer forms an output probability score to determine whether the sample is real or not using the sigmoid function. Besides that, the loss function of the discriminator is given by:

$$L_{\mathrm{D}} = -\mathbb{E}[\log D(x, y)] - \mathbb{E}[\log(1 - D(G(z, y), y))] \tag{4}$$

where real samples of the dataset are denoted by x, and the generated samples by $G(z, y)$. This means that the discriminator is trained to discover the real samples and most distinctly distinguish them from the synthetic ones. Figure 2 illustrates the architecture of the CGAN used for synthetic attack data generation.

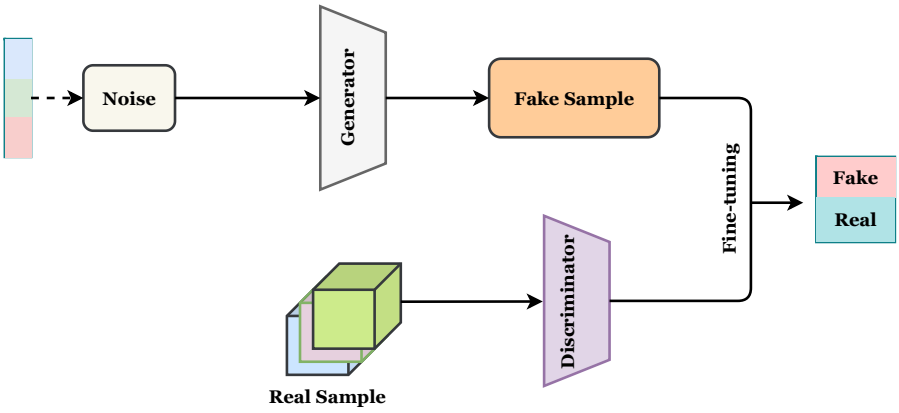

Fig. 2: GAN Architecture for Synthetic Data Generation

The training of CGAN uses an iterative manner in which the generator and discriminator are alternately updated. First, L_{D} is optimized on the discriminator to

ensure that it correctly classifies real images and punishes generated images. Then, the generator is modified to minimize L_{G}, making it more capable of producing realistic attack samples that can defeat the discriminator. Algorithm 1 gives an overview of the whole adversarial training process.

Algorithm 1 CGAN Training Process

Require: Real data samples $x \sim p_{data}(x)$, noise vector $z \sim p_z(z)$, class labels $y \sim p_y(y)$, generator G, discriminator D, learning rates α_G, α_D, number of iterations N.

Ensure: Trained generator G^* producing realistic attack samples.

0: **for** $i = 1$ to N **do**

0: Sample minibatch of real data $(x, y) \sim p_{data}(x, y)$

0: Sample noise vector $z \sim p_z(z)$

0: Generate synthetic samples:

$$\tilde{x} = G(z, y)$$

0: Compute discriminator loss:

$$L_D = -\mathbb{E}_{(x,y)\sim p_{data}}[\log D(x, y)] - \mathbb{E}_{(z,y)\sim p_z}[\log(1 - D(G(z, y), y))]$$

0: Update discriminator:

$$\theta_D \leftarrow \theta_D + \alpha_D \nabla_{\theta_D} L_D$$

0: Compute generator loss:

$$L_G = -\mathbb{E}_{(z,y)\sim p_z}[\log D(G(z, y), y)]$$

0: Update generator:

$$\theta_G \leftarrow \theta_G - \alpha_G \nabla_{\theta_G} L_G$$

0: **end for**

0: **Return:** G^*

=0

3.4 Intrusion Detection Using Transformer-Based IDS

The Transformer-based IDS has a hierarchical architecture including numerous components that process network traffic data in a systematic manner and classify it as normal or attack. To compensate for the high dimensionality of network traffic features and to maintain computational efficiency, the model architecture has been designed. The architecture has the Input Layer, Transformer Encoder, and fully connected Layers.

The transformer encoder, as part of the IDS model, is the core part where 6 stacked layers, with each layer consisting of multi-head self-attention and a position-wise feed-forward network. Unlike recurrent architectures that process sequential data in a step-wise manner, transformers use self-attention to learn how important each feature is compared to the rest of the data when performing association analysis

on network traffic attributes. For example, in each of the encoder layers, the model has an attention mechanism called a multi-head self-attention mechanism that has 8 attention heads that attend to various aspects of the data at the same time. After passing the output of the self-attention layer through a feed-forward net with ReLU activation, they are further refined before being propagated to the subsequent layers. Mathematically, the transformation at each encoder layer can be denoted as:

$$Z = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{5}$$

where Q, K, and V are the query, key, and value matrices produced from the input data, respectively; and d_k denotes the dimensionality of each attention head. This is used so that the layer normalization is applied with each step to stabilize training and mitigate vanishing gradient problems. Details of the hyperparameters used for training the proposed model are summarized in Table 1.

Table 1: Hyperparameters of Transformer-based IDS During Training

Parameter	**Value**
Loss Function	Categorical Cross-Entropy
Optimizer	AdamW
Learning Rate	0.0001
Batch Size	128
Epochs	50
Dropout Rate	0.3

The feature representation that has gone through the Transformer Encoder is subsequently classified by fully connected layers. A dense layer with softmax activation is finally used on the output layer that produces a probability score for each class label which helps the detection of multi-class attacks. The final classification decision is based on:

$$P(y_i) = \frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}} \tag{6}$$

where $P(y_i)$ is the probability of class i and z_i is the logits based on the final dense layer. Figure 3 shows the overall architecture of the transformer-based IDS.

4 Results & Discussion

The transformer-based IDS was trained and tested with the TON_IoT dataset that contains a large amount of attacks present. The model was evaluated on a separate test set, on separated training (not shared with validation) and validation phases. The classification performance of our proposed transformer-based IDS and the impact of generative AI in cloud security are discussed in this section.

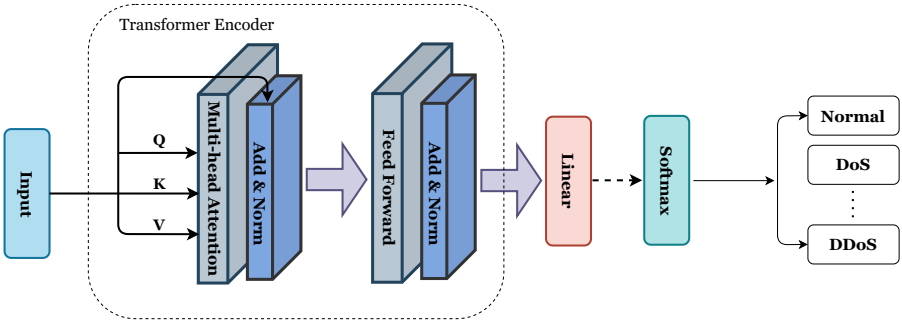

Fig. 3: Transformer-based IDS Model Architecture

4.1 Classification Performance

The classification performance of the Transformer-based IDS is based in Table 2 across 10 different network traffic classes, including normal traffic and different cyberattacks. Results show that the model is able to classify between benign and malicious network activity as real-time and accurate cloud security enricher.

Table 2: Performance Metrics of Transformer-Based IDS For Different Classes

Attack Type	Precision	Recall	F1-Score	Accuracy
Normal	98.2%	98.5%	98.3%	98.4%
Scanning	96.7%	96.1%	96.4%	96.5%
Injection	94.5%	93.8%	94.1%	94.3%
DoS	97.3%	96.9%	97.1%	97.2%
DDoS	98.5%	97.8%	98.1%	98.2%
Password	95.6%	94.9%	95.2%	95.3%
Ransomware	96.2%	95.5%	95.8%	96.0%
MITM	93.9%	93.1%	93.5%	93.7%
XSS	94.8%	94.2%	94.5%	94.6%
Backdoor	97.1%	96.8%	96.9%	97.0%
Overall	**96.7%**	**96.0%**	**96.3%**	**96.5%**

The overall accuracy of 96.5% shows that the IDS is very reliable in detecting cyber threats in the cloud environment. It is worth mentioning that the detected accuracy is highest for both DDoS and normal traffic, with precision values of 98.5% and 98.2% respectively indicating the model's robustness in identifying large-scale DDoS attacks. Specifically, DDoS incursions are highly detrimental in cloud computing as DDoS incursions can strain virtualized infrastructure, reduce service availability, and result in downtime. The system enhances the cloud service's resilience by effectively detecting DDoS traffic.

Notably, the detection accuracy ends up being higher among DDoS and normal traffic with precision values of 98.5% and 98.2% respectively, indicating the model is very robust for detecting large-scale DDoS attacks. In particular, DDoS attacks are

highly important in cloud computing due to the fact that DDoS incursions can crash virtualized infrastructure, degrade service availability, and cause downtime. Through effective DDoS traffic detection, the system prevents losses caused by the lack of resilience in the cloud service.

Moreover, the model is able to effectively classify sophisticated attacks such as backdoor intrusions and ransomware with F1-scores of 95.8% and 96.9%, respectively. Cloud security is vulnerable to ransomware attacks that infest its server with encryption that chokes critical data and then demands a ransom, while backdoor attacks give adversaries direct access to the server. Accurate classification of these threat attacks enables the IDS to proactively mitigate threats in the cloud such that a breach or unauthorized access to data is minimized.

At the same time, some attack types including MITM and injection attacks have recall values of 93.1% and 93.8% respectively, which means that they have admitted some misclassified samples. In this case, it indicates that the detection would benefit from additional model optimization such as additional synthetic samples made by the CGAN. In particular, injection attacks are injected into cloud databases and web services, which make detection of them necessary in order to prevent data corruption and exfiltration.

Figure 4 further shows the accuracy, precision, recall, and F1-score of each class as a comparison of the performance of our IDS in dealing with different types of attacks. This visualization helps to understand that the model does a good job of discriminating normal traffic against attack types, especially considering highly severe threats such as ransomware, DDoS, and injection attacks. The balanced and relatively high performance across all categories demonstrates the robustness of a transformer-based IDS in protecting the cloud environment against various intrusion attempts.

4.2 Impact of CGAN-Based Synthetic Data Generation

To evaluate the quality of the generated attack samples, we projected high-dimensional network traffic data using t-SNE in a lower-dimensional space. Figure 5 is a visual representation of real vs synthetic attack data distribution, which makes it easy to evaluate how synthetic samples are close to real attack patterns.

As shown in Figure 5, the t-SNE has a high overlap between the real and the synthetic attack data set clusters. The high level of correspondence between the synthetic and real attack behaviors indicates that the CGAN was capable of learning the true data distribution and generating synthetic samples that more or less mimic real attack behaviors. Therefore, IDS models are expected to be better trained on this enriched dataset, as detection accuracy for rare or less frequently used attack types is expected to improve.

We also investigated how much synthetic data augmentation increases (Enhanced IDS) as compared to no synthetic data augmentation (Baseline IDS) to quantify the impact of CGAN-based data augmentation on the Transformer-based IDS model. As shown in Table 3, the results are summarized in terms of a performance report on key evaluation metrics such as accuracy, precision, recall, and F1-score.

From the table, it can be seen that CGAN generated samples bring a significant improvement in all the performance metrics. The classification performance was

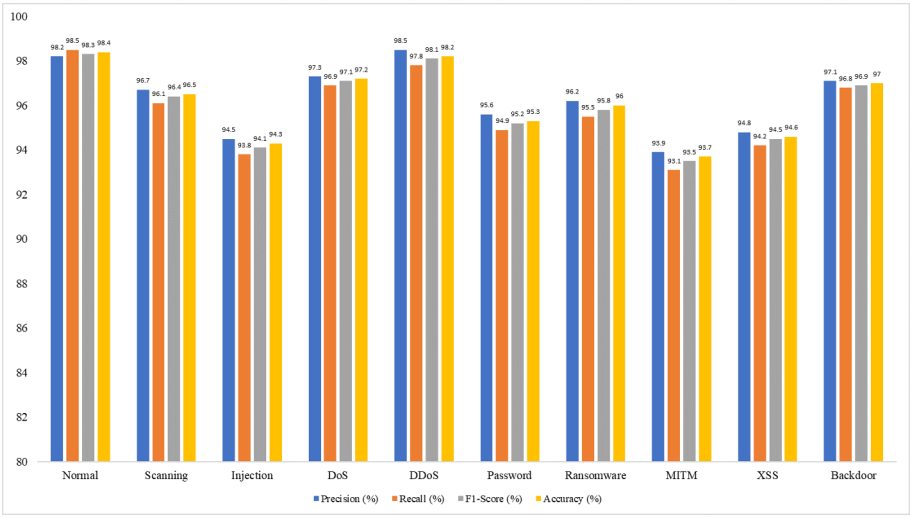

Fig. 4: Performance Comparison of Transformer-Based IDS Across Different Attack Types

Table 3: Impact of CGAN-Based Data Augmentation on IDS Performance

Metric	Without CGAN	With CGAN
Accuracy	93.4%	**96.5%**
Precision	92.1%	**96.7%**
Recall	91.8%	**96.0%**
F1-Score	91.9%	**96.3%**

improved by increasing the accuracy of the IDS from 93.4% to 96.5%. Likewise, precision, recall, and F1-score significantly improved, which shows that the model can now take better notice of a larger variety of attack classes. Specifically, it is useful for cloud security as it significantly improves the security posture of the cloud detection system which should be more robust and resilient against a variety of and ever-evolving cyber threats.

The augmentation by CGAN extends the generalizability of the IDS in that it produces realistic attack variations so that the IDS can be more effective in detecting known as well as unknown attacks. This approach provides a drastic improvement in intrusion detection in a cloud environment where the cyber threats are evolving at a high speed and traditional detection models are not updated accordingly.

4.3 Comparative Analysis

We demonstrated the effectiveness of our Transformer based IDS with CGAN augmentation using both existing intrusion detection models applied on different datasets.

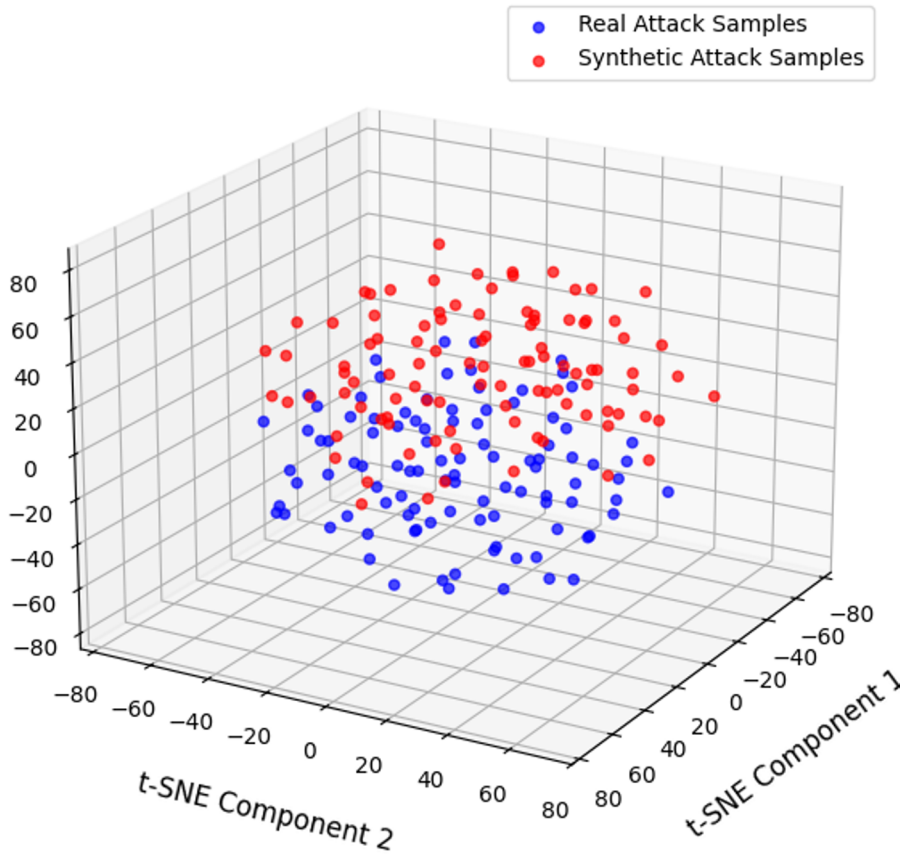

Fig. 5: 3D t-SNE Visualization of Real vs. Synthetic Attack Samples

This comparative analysis is presented in Table 4, which demonstrates how our approach is much better than traditional machine learning models.

Table 4: Comparison of Models on Different Datasets

Reference	Dataset	Model	Accuracy
[14]	NSL-KDD Dataset	AE	87%
[?]	NSL-KDD Dataset	SVM	75.4%
[15]	Kitsune Network Attack Dataset	CVAE	78.61%
Ours	ToN_IoT Dataset	TE	**96.5%**

Our model outperformed the AE model introduced by Ieracitano et al. [14] on the NSL-KDD dataset, obtaining 9.5% higher accuracy than their model, which could

reach 87% accuracy. Like the Support Vector Machine (SVM) impression created in that approach, which was developed by Park et al. [**?**] on NSL-KDD and attained just 75.4% accuracy, our model clearly outperformed that one by 21.1%. In addition, the conditional Variational Autoencoder model proposed by Aceto et al. [15] on the Kitsune Network Attack dataset had an accuracy of 78.61%, from which our model was 17.89% higher.

On the ToN_IoT dataset, our Transformer IDS trained on CGAN-generated synthetic data outperformed all preexisting models with an accuracy of 96.5%, achieved is an outstanding 96.5%. The improvement is remarkable and can be explained by the rich feature extraction that Transformers bring along with the diversity attributed to CGAN-generated attack samples. However, our model's learning was greatly enhanced by Generative AI because Generative AI can help the model generalize better to real-world cyber threats in the cloud.

5 Conclusion

The work suggested a Generative AI-enhanced IDS based on Transformer-based anomaly detection and CGANs to improve cloud security. Often reducing standard IDSs, the approach efficiently solves important cybersecurity difficulties such as data imbalance, adversarial evasion, and real-time threat detection. With a 96.5% detection accuracy on the ToN-IoT dataset, GAI-IDS achieves great adaptability and precision by creating synthetic attack samples, boosting underrepresented threat classes, and using multi-head self-attention techniques, thereby greatly surpassing standard IDS models. Furthermore, adversarial training methods strengthen the system against complex evasion strategies, hence increasing its resilience to new cyberattacks. Although GAI-IDS shows impressive progress, further research is required to guarantee its scalability, efficiency, and real-world applicability. Future studies should concentrate on adaptive learning systems that enable continuous updates free from retraining, so improving the model's capacity to dynamically behave to change attack patterns.

References

[1] K. Khanna, Enhancing cloud security with generative ai: Emerging strategies and applications, Technology (JARET) 3 (1) (2024) 234–244.

[2] I. Martins, J. S. Resende, P. R. Sousa, S. Silva, L. Antunes, J. Gama, Host-based ids: A review and open issues of an anomaly detection system in iot, Future Generation Computer Systems 133 (2022) 95–113.

[3] N. N. I. Prova, Enhancing fish disease classification in bangladeshi aquaculture through transfer learning, and lime interpretability techniques, in: 2024 4th International Conference on Sustainable Expert Systems (ICSES), IEEE, 2024, pp. 1157–1163.

[4] S. Bengesi, H. El-Sayed, M. K. Sarker, Y. Houkpati, J. Irungu, T. Oladunni, Advancements in generative ai: A comprehensive review of gans, gpt, autoencoders, diffusion model, and transformers., IEEe Access (2024).

[5] A. Yaseen, Uncovering evidence of attacker behavior on the network, ResearchBerg Review of Science and Technology 3 (1) (2020) 131–154.

[6] N. N. I. Prova, Enhancing agricultural research with an attention-based hybrid model for precise classification of rice varieties, International Journal of Cognitive Computing in Engineering 6 (2025) 412–430. doi:10.1016/j.ijcce.2025.02.002.

[7] G. S. Nadella, S. R. Addula, A. R. Yadulla, G. S. Sajja, M. Meesala, M. H. Maturi, K. Meduri, H. Gonaygunta, Generative ai-enhanced cybersecurity framework for enterprise data privacy management, Computers 14 (2) (2025) 55.

[8] A. Dunmore, J. Jang-Jaccard, F. Sabrina, J. Kwak, A comprehensive survey of generative adversarial networks (gans) in cybersecurity intrusion detection, IEEE Access 11 (2023) 76071–76094.

[9] J. L. López Delgado, J. A. López Ramos, A comprehensive survey on generative ai solutions in iot security, Electronics 13 (24) (2024) 4965.

[10] U. Kumaran, S. Thangam, T. Prabhakar, J. Selvaganesan, H. Vishwas, Adversarial defense: A gan-if based cyber-security model for intrusion detection in software piracy, Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications 14 (4) (2023) 96–114.

[11] M. Andreoni, W. T. Lunardi, G. Lawton, S. Thakkar, Enhancing autonomous system security and resilience with generative ai: A comprehensive survey, IEEE Access (2024).

[12] T.-H. Vu, S. K. Jagatheesaperumal, M.-D. Nguyen, N. Van Huynh, S. Kim, Q.-V. Pham, Applications of generative ai (gai) for mobile and wireless networking: A survey, IEEE Internet of Things Journal (2024).

[13] N. N. I. Prova, Healthcare fraud detection using machine learning, in: 2024 Second International Conference on Intelligent Cyber Physical Systems and Internet of Things (ICoICI), IEEE, 2024, pp. 1119–1123.

[14] C. Ieracitano, A. Adeel, F. C. Morabito, A. Hussain, A novel statistical analysis and autoencoder driven intelligent intrusion detection approach, Neurocomputing 387 (2020) 51–62.

[15] G. Aceto, F. Giampaolo, C. Guida, S. Izzo, A. Pescapè, F. Piccialli, E. Prezioso, Synthetic and privacy-preserving traffic trace generation using generative ai models for training network intrusion detection systems, Journal of Network and Computer Applications 229 (2024) 103926.