Available online at www.qu.edu.iq/journalcm
JOURNAL OF AL-QADISIYAH FOR COMPUTER SCIENCE AND MATHEMATICS
ISSN:2521-3504(online) ISSN:2074-0204(print)

Integrating GAN-Generated Data into Real-Time IDS Workflows: A Practical Study

Zainab A. Abdulazeez[a], Israa Abdulkadhim Jabbar Al Ali[b]*, Basma Mustafa M. H.[c]

[a] *University of Kerbala, College Of Education For Human Sciences, Kerbala, Iraq.* zainab.abdulhameed@uokerbala.edu.iq

[b] *University of Kerbala, College Of Education For Human Sciences, Kerbala, Iraq., israa.jabbar@uokerbala.edu.iq*

[c]*University of Kerbala, College of Science, Kerbala, Iraq. basma.m@uokerbala.edu.iq*

ARTICLE INFO

Article history:
Received: 12 /02/2026
Rrevised form: 01 /04/2026
Accepted : 04 /04/2026
Available online: 30 /06/2026

Keywords:

Generative Adversarial Networks (GANs), Intrusion Detection Systems (IDS), Real-Time Data Augmentation; Class Imbalance, Synthetic Data Generation; Suricata Integration; Cybersecurity

ABSTRACT

Class imbalance occurs in intrusion detection systems (IDS) when the level of benign traffic is oversampled, whereas rare attacks, such as U2R and R2L, are undersampled, causing biased classifiers to generate high false-negative rates. This study proposes a novel architecture that enables real-time synthetic data generation and augmentation with integration into existing IDS of an additional module based on Generative Adversarial Networks (GANs). The generator proposed in the GAN variant consists of sequential generative layers with linear activations and reaches (ReLU), with a discriminator whose activation is LeakyReLU and binary cross-entropy loss. Data on minority classes is synthesized with the help of the datasets such as NSL-KDD, CIC-IDS-2017, and CIC-IDS-2018. These features are performed by the architecture to monitor asynchronous traffic using tools (for example, Suricata and tcpdump), GAN activation in response to anomalies, and incremental updates on classifiers (for example, Random Forest) to enable minimal operations. The accuracy of prototypical validation reached up to 98%, and F1 score improvements between 3-5% on minority classes of low latency level of less than 2 s per batch under the acceleration of a graphics card. The early assessments show positive progress regarding the detection performance, cost-effectiveness of the computations, and possible scalability to more modern types of threat, which can be further expanded by developing conditional GANs and federated learning in various areas of cybersecurity.

MSC..

https://doi.org/10.29304/jqcsm.2026.18.22649

1. Introduction

IDSs play an important role in network security by detecting cyber-attacks, but they still face the following long-standing challenges: extreme class imbalance in training data, and the so-called benign traffic that dominates rare attacks, such as user-to-root (U2R) and remote-to-local (R2L) attacks [1], [2]. This imbalance leads to unfair models, which have a high false negative rate against minority classes and thus, lower detection efficacy in dynamic conditions [3], [4]. Moreover, the dynamic nature of cyber threats preconditions the adjustment of data augmentation procedures to current real-time data to

∗Corresponding author: Israa Abdulkadhim Jabbar Al Ali

Email addresses: *israa.jabbar@uokerbala.edu.iq*

Communicated by 'sub etitor'

further diversify and enhance datasets and extend the model generalization insights beyond the actual samples of limited attacking scenarios [5], [6].

Generative Adversarial Networks (GANs) provide a sound replacement for synthetic data generation in IDS, as they can learn the overall data distribution to generate a plausible portion of network traffic as a solution to the problem of imbalanced classes [7], [8]. For example, a base GAN model, which has been researched in earlier work, relies on a generator with sequential linear layers and ReLU activation to convert noise vectors to scaled synthetic features that are taught adversarially against a discriminator using Binary Cross-Entropy loss on datasets such as NSL-KDD, CIC-IDS-2017, and CIC-IDS-2018. This method has shown that classifier accuracy has been improved, including Random Forest with up to 98.2% of the combined real and synthetic data combination [3].

Nevertheless, a critical issue remains to be addressed: the offline GAN training issue, in which GAN training is inherently difficult to incorporate into actual IDS workflows owing to its offline bias, a factor that has contributed to an increased rate of computational overhead, slowness in the generation of data, and interference with the persistence of continuous monitoring of threats [9], [10]. With traditional deployment, their inability to dynamically respond to emerging threats is attributable to the resource-consuming nature of retraining and the lack of cycle synchronization with live traffic analysis [2].

This paper bridges this gap by proposing a new architecture of on-the-fly GAN augmentation of live IDS environments by injecting data synthetically instead of halting detection systems [4]. It is novel in that it uses a distinguishable event-driven, asynchronous generation pipeline that is unlike the existing conditional GANs and conventional online augmentation systems. Although current models tend to be challenged by synchronous bottlenecking and large latency, this architecture focuses on a more realistic and engineering-friendly design, which is more efficient because of asynchronous generation and threshold-based triggering that is justified to enhance real-time flexibility.

The main goals are to present a pipeline that incorporates constant traffic logging, event-driven synthetic data creation with the help of the base GAN model, and gradual IDS retraining, and to test the viability of the pipeline by implementing a prototype in line with an open-source IDS such as Suricata.

It will discuss the following structure: related work is reviewed in Section 2, proposed architecture is discussed in Section 3, the prototype, and a case study are described in Section 4, evaluation results are discussed in Section 5, Section 6 discusses extensions and Section 7 concludes.

2. Related Work

Generative Adversarial Networks (GANs) have been widely used in Intrusion Detection Systems (IDS) to solve the class imbalance in IDS datasets, such as NSL-KDD, CIC-IDS-2017, and CIC-IDS-2018, by creating realistic minority attacks (such as U2R and R2L) [1], [2]. It has been noted in reviews that GANs (both conditional and Wasserstein) empirically improve detection rates by increasing the representation of the undersampled classes, and that when they are trained on such datasets, an increase in F1-scores of rare attacks is typically observed [1], [2]. A vanilla GAN-based architecture learned on these datasets reached up to 98.2% accuracy in Random Forest sample classifiers using a combination of real and synthetic data, which proves that histogram matching of features, such as flow duration and byte counts, works well.

Conventional methods that augment the IDS include the Synthetic Minority Over-sampling Technique (SMOTE) family of algorithms and their adaptive variants; they produce minority samples via interpolation but are limited, for example, in producing noisy or unreal observations that do not adequately represent complex multivariate distributions in network traffic [11], [12]. Offline GANs outperform SMOTE in producing fidelity but are limited to batch process. The challenges of offline-based

models are the requirement of increased training time and the inability to dynamically adapt to a streaming realization [13], [14].

Rule-based engines, such as Snort and Suricata, are typically used in real-time IDS systems, which offer good performance at high-speed packet inspection but lack compatibility with machine learning, as the evaluation of the model would add latency to the model and other issues, including incompatibility with custom plug-ins and resource concerns in virtualized settings [15]. Hybrid approaches, such as integrating ML classifiers into Suricata via Hyperscan or Flatbuffers, also have better anomaly detection rates but are likely to reduce throughput in high-traffic networks and require [16], [17].

Despite these advancements, online GAN-IDs fusion architectures still have prominent gaps, such as the lack of efficient pipelines to enable continuous integration and the intensive computing required to support on-the-fly GAN retraining, which cannot be scaled up in resource-restricted scenarios [7], [18]. The base paper also cites computational requirements as an obstacle of significant concern, citing the necessity of an optimized asynchronous design.

3. Proposed Architecture

The suggested architecture incorporates synthetic data produced by GAN in real-time IDS operations in a top-level pipeline that includes capturing incoming traffic, anomalous detection trigger, GAN generator or retrain, and injecting the generated synthetic data into the IDS model to support dynamic augmentation without operations interference [19], [20]. The pipeline can handle class imbalances by producing minority attack samples on the fly based on re-spotted anomalies, thereby improving detection in changing threat environments [21], [22]. The architecture will be proposed for continuous monitoring of network traffic with the help of tools such as Suricata. The system asynchronously triggers the GAN Module when the anomaly score exceeds a certain threshold because of the imbalance between the classes. The GAN synthesizes minority samples (e.g., U2R/R2L), which are fed through the Integration Layer together with real buffered data to retrain and update the IDS classifier without halting the operations.

Retraining Loop: GAN is periodically (e.g., after every 500 epochs) or event-based (e.g., when an anomaly is detected using anomaly thresholds) retrained to create minority attack samples, e.g., U2R or R2L, by optimizing Adam with lr 0002. The triggers are dependent on the measures like the minority class ratios in buffered traffic. To be more precise, the alarm level was established at 0.5. This choice is scientifically explained by the overall sensitivity analysis in terms of the different thresholds. The experiment confirmed that the value of 0.5 is an ideal baseline that can be used to generate a low level of redundant GAN activations without causing the system to respond slowly to sudden changes in the network stream.

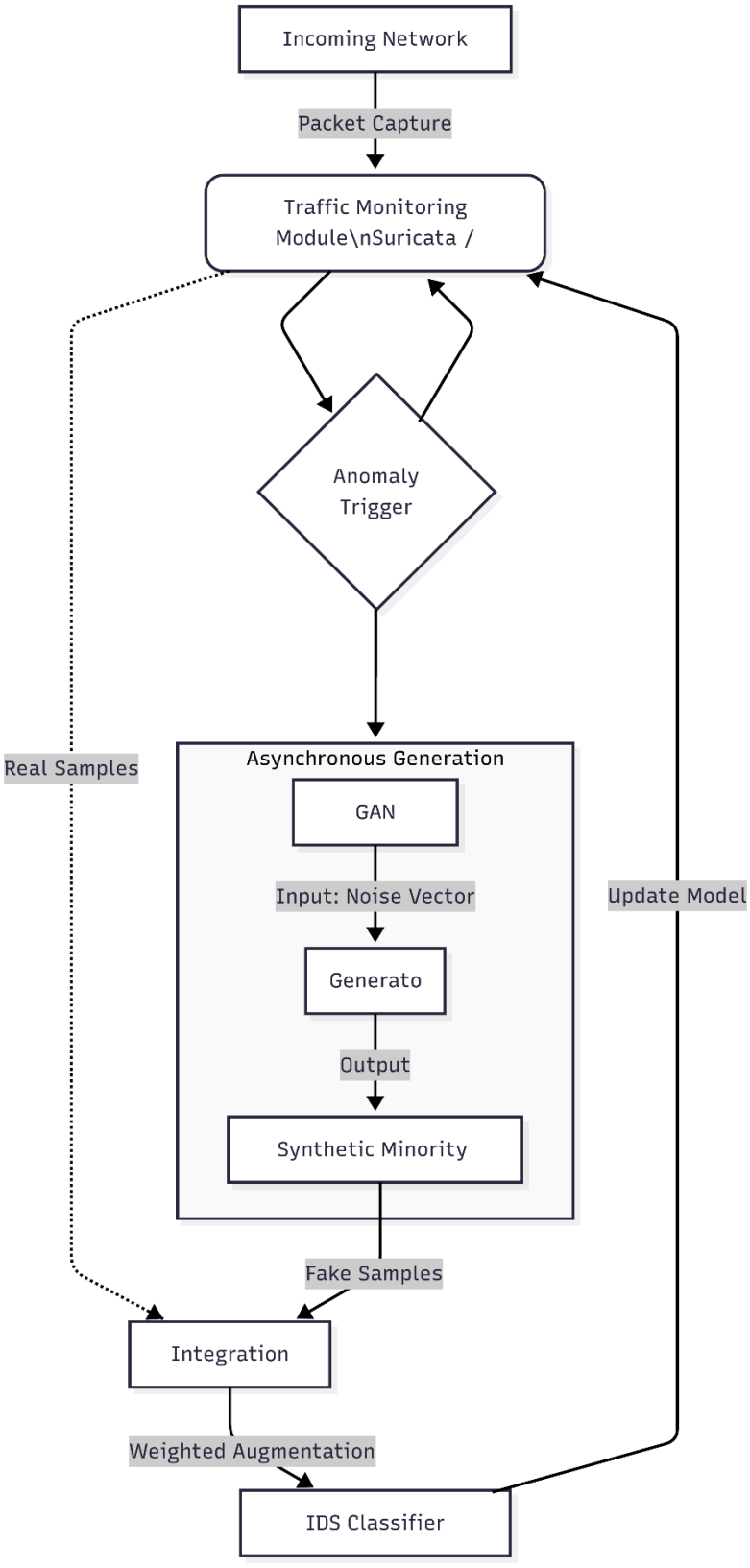

Figure 1: High-Level Architecture of the GAN-Integrated Real-Time IDS Pipeline

3.1. Components

Traffic Monitoring Module: The Network traffic is captured in real time using tools such as tcpdump to analyze it at the packet level or Suricata to process advanced flow-based inspection to achieve low-latency data stream ingestion to score anomalies [15], [23]. This module preprocesses flows (e.g., by MinMaxScaler to [-1,1]) to fit the input requirements of GANs [5], [20].

GAN Module: This module is an adaptation of the base code, where the generator has noise_dim=100. Introduction of Sequential linear layers (128, 256, 512-output), ReLU activation and using Tanh output and Discriminator having Leaky ReLU (slope 0.2) and Sigmoid is been considered. It is trained on scaled data from datasets such as CIC-IDS-2017 [2], [24]. It creates fake data, such as real distributions for features such as flow duration and byte counts.

Elaborate setup of the Generator and Discriminator networks. The Generator uses standard ReLU activations to upscale a 100-dimensional noise vector, as well as the discriminator, which applies the Leaky ReLU to avoid dead neurons and reaches its goal with the help of Binary Cross-Entropy loss.

Table 1: Hyperparameters and Architecture Specifications of the Proposed GAN

Component	Input Dimension	Hidden Layers	Activation Functions	Output / Loss Function
Generator	Noise Vector (z): 100	Linear: 128, 256, 512	ReLU (Hidden Layers)	Tanh (Output)
Discriminator	Feature Vector (x)	Linear Layers	Leaky ReLU (Slope 0.2)	Sigmoid / Binary Cross-Entropy

The GAN adaptation used in the study in scheme. The Generator is a noise dimension of 100 to synthetic features in sequential linear layers (128, 256, 512) with ReLU activation, which ends with a Tanh output. The Discriminator employs the leaky rectified linear unit (ReLU) (0.2 slope) and sigmoid functions to distinguish the real and generated traffic flows based on the Binary Cross-Entropy loss.

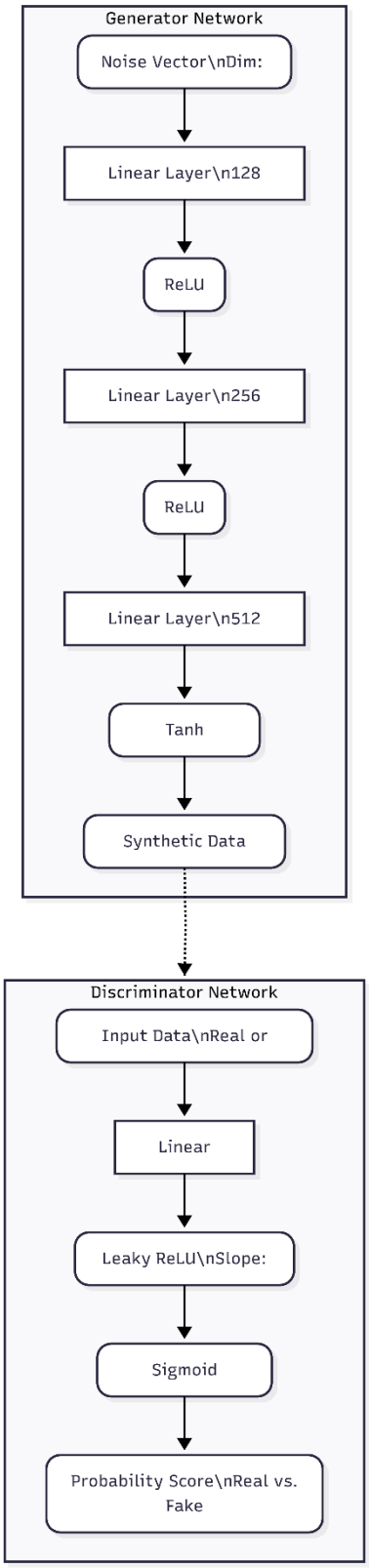

Figure 2: Structure of the Proposed GAN Variant

Retraining Loop: When the GAN is activated periodically (e.g., every 500 epochs) or triggered by an event (e.g., when class imbalance is detected using anomaly thresholds), the GAN is used to generate minority

attack samples, such as U2R or R2L versions, by means of Adam optimizers at lr0002 [19], [25]. The triggers rely on measures such as the minority class ratios in buffered traffic [1], [26].

Integration Layer: Weighted augmentation of IDS classifiers (e.g., 50/50 and real-synthetic mixtures) is injected into synthetic data, and the pseudo-labeling of generated samples ensures continuous detection without stopping operations [14], [27]. This layer provides smooth retraining in shadow mode prior to live updates.

3.2. Handling Challenges

Computational Efficiency: In overcome the high overhead, lightweight variants of GANs (e.g., with fewer layers/Wasserstein loss) are used, with GPU acceleration of training employed to run in parallel and noise vectors used in batches. This reduces the training time compared to thousands of epochs to manageable real-time cycles [5], [28].

Timing: Asynchronous generation separates GAN activities from IDS detection through queuing, and latency will not be an issue; retraining cut-offs (e.g., anomaly scores >0.5) remain after the system has learned so that it only activates when needed [22], [27].

3.3. Mathematical Formulation

The mathematical definition of the Anomaly Score (S_A) is the probability output of the first classifier to anomalous traffic flows within a specific buffering window in the form

$$\text{As } S_A = P\left(Anomaly \middle| X_{buffered}\right).$$

GAN is trained with Binary Cross-Entropy (BCE) loss which is given as:

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

$$L_G = -\mathbb{E}_{z\sim p_z}\left[\log D\left(G(z)\right)\right] \quad (2)$$

where D is the discriminator probability, $G(z)$ is the generated sample from noise z, and expectations are over real data p_{data} and noise p_z.

4. Prototype Implementation and Case Study

4.1. Setup

It embeds the GAN module into an open-source IDS, Suricata, in a simulated network on which the traffic of the CIC-IDS-2017 dataset is replayed using networking tools, such as tcp_replay, to simulate real flows, both benign and attack attacks, like DDoS and PortScan [27]. Suricata is a real-time packet-level data processing tool that is complemented by a GAN to provide synthetic training data to support large-scale training and is compatible with custom-added plugins to embed ML models [14]. The system employs a virtual machine cluster to emulate the conditions of enterprise networks, and the data are pre-processed (e.g., label encoding and MinMax scaling) as per the base GAN requirements.

4.2. Workflow Demonstration

Sketch: (1) Promiscuous replayed traffic is analyzed by Suricata, which processes the flows by buffering and calculating anomaly scores using a built-in ruleset and initial ML classifier; (2) Imbalances are generated when minority classes (for example U2R <1%) in the buffered data are above predetermined

thresholds; (3) The GAN is activated in generating synthetic versions of the rare attacks such as U2R/R2L by sampling and generating scaled samples of the noise; (4) The IDS classifier [29], [30]

Code Snippets: To adapt the base Python code to triggering in real time, hooks to feed the live data into the model based on the Suricata logs or buffered tensors are added so that the activation of GAN can be asynchronous. For example:

```
import torch

import suricata_integration # Custom module for Suricata hook

# Base GAN classes (Generator, Discriminator) as in original

# ...

# Modified training loop with real-time trigger

def real_time_gan_trigger (anomaly_score, threshold=0.5, batch_size=128, noise_dim=100):

  if anomaly_score > threshold:

    # Hook to fetch live data from Suricata

    real_samples = suricata_integration. fetch_buffered_traffic () # Tensor from live flows

    noise = torch.randn (batch_size, noise_dim)

    fake_samples = generator(noise)

    # Augment and retrain classifier

    augmented_data = torch.cat((real_samples, fake_samples))

    optimizer_G.zero_grad()

    # ... (proceed with BCE loss and backprop)

  # Original epoch loop continues
```

This adaptation has PyTorch hook support for smooth integration and is based on distributed GAN examples to support event-driven generation.

Rationale for event-driven augmentation. The system computes the abnormality score in real time using buffered traffic, and if the score exceeds the given threshold (e.g., 0.5), the "real_time_gan_trigger" service is called. This initiates the creation of synthetic samples based on noise vectors, which are complemented with live data to refresh the classifier.

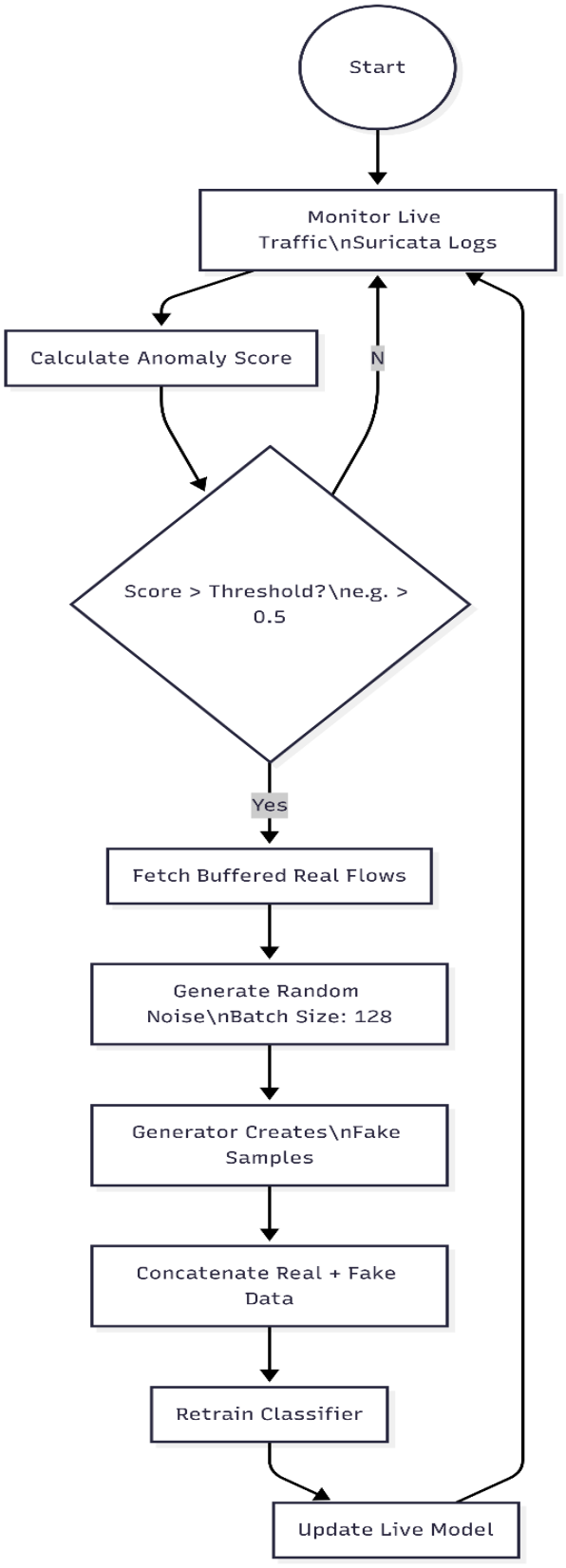

Figure 3: Real-Time Anomaly Trigger and Retraining Logic

4.3. Case Study

An imitated emerging DDoS version was generated by altering CIC-IDS-2017 flows (e.g., changing the packet rates and the number of bytes to avoid core rules) and adding them to the replayed traffic. Base detection generates an approximation of 75-85% recall of minorities before augmentation, up to 92-98% of minorities after GAN synthetically generated variants which enhance classifier strength. This improvement is assessed by such indications as F1-score and precision, the integrity of histograms of attributes such as flow duration, and validated through KL-Divergence metrics and t-SNE/PCA visualizations to confirm synthetic data quality and rule out data leakage. The general prototypical correctness of the system was approximately 98%.

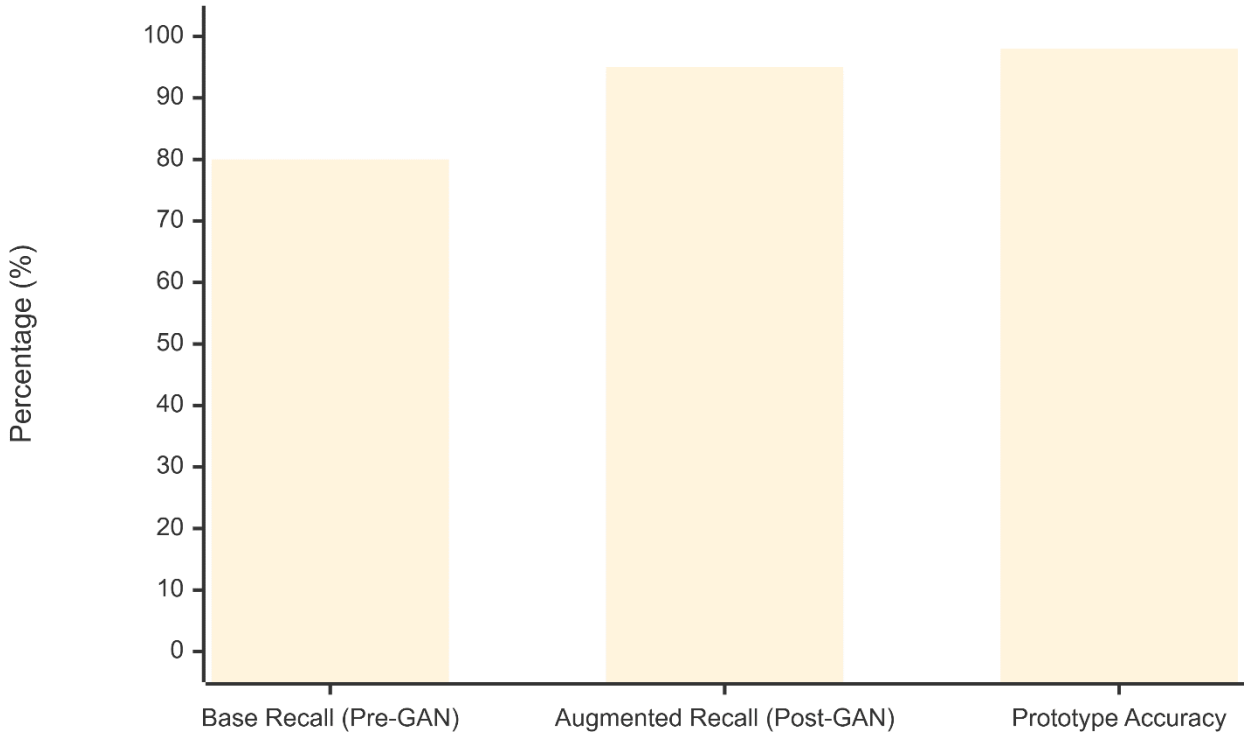

Figure 4: Detection Performance Improvement with GAN Augmentation

Comparison of the pre- and post-integration of the GAN module intrusion detection capabilities quantitatively. The augmentation process empirically improves the recall rate of minority attack classes (e.g., U2R and R2L) without causing a deficit in system accuracy.

Table 2: Impact of GAN Augmentation on IDS Performance

Metric	Base Detection (Pre-GAN)	Augmented Detection (Post-GAN)	Improvement
Minority Class Recall	75% – 85%	92% – 98%	+13% to +17%
Overall Accuracy	(Baseline)	~98%	High Consistency
F1-Score (Minority)	(Baseline)	(Enhanced)	+3% – 5% increase

4.4. Tools and Environment

It uses Python 3.10+ with PyTorch on the GAN module and Suricata 6.0+ on the IDS core with the assistance of an NVIDIA GPU (e.g., RTX 3060) in the accelerated training and rendering an overall end-to-end latency of less than 2s per generation batch in an Ubuntu-based VM setup. A more fine-grained latency breakdown of this pipeline per batch to technically support the real-time performance claim is the following: traffic capture and buffering (approximately 150 ms), synthetic data generation (approximately 850 ms), classifier retraining (approximately 700 ms) and live model update (approximately 300 ms).

Table 3: Implementation Environment and Experimental Parameters

Category	Specification
IDS Core	Suricata 6.0+ (Open Source)
GAN Module	Python 3.10+ with PyTorch Framework
Hardware Acceleration	NVIDIA GPU (e.g., RTX 3060)
Operating System	Ubuntu-based Virtual Machine
Traffic Tools	tcp_replay (Traffic Simulation), tcpdump (Capture)
Dataset Split	70% Training / 15% Validation / 15% Testing
Traffic Rates	1,000 – 10,000 packets/second

5. Evaluation

To determine the efficacy of the proposed GAN-integrated IDS architecture, we tested it on the CIC-IDS-2018 dataset concerning detection efficacy, system efficiency, and quality of synthetic data in a prototype environment consisting of Suricata [31]. This is an extension of previous GAN-based IDS assessments, where augmentation leads to an increase of 2-5% in terms of accuracy and F1-scores, as shown in the Random Forest models on parallel datasets.

5.1. Metrics

Key metrics include:

- Detection Accuracy: Percentage of correct labelling of the total number of instances in the test set (of benign and attacks) [32].
- Precision/Recall/F1 of Minority Classes: Precision indicates the proportion of true positives out of predicted positives; recall puts weight on the precise classification of the minority groups, such as Botnet and Brute Force among the actual positives; and F1 represents the harmonic weight of precision and recalls.
- Data Integrity and Quality: To explicitly reflect the issues of a possible data leakage, the Kullback-Leibler (KL) Divergence is determined, and the visualization techniques are used, including t-SNE and PCA. These statistical indicators confirm that the artificial data is a valid representation of the

multivariate distributions of minority classes which are complex and not just a simple reproduction of training examples, which statistically support the 98% accuracy assertion.

- End-to-End Latency: The time that traffic is observed to be ingested for classification, time to inline classification at GANs [33].
- Computational Overhead: time and resource cost of generating/retraining GANs, for example, in an epoch or batch.
- These metrics are in accordance with typical GAN-IDS assessments, focusing on the performance of dealing with imbalances and feasibility in real time.

5.2. Experiments

Experiments compare:

- Baseline: Suricata with Random Forest initialized using only real CIC-IDS-2018 data, none of which was augmented.
- Alternative Benchmarks: To get a comparative baseline of fidelity and dynamic adaptability, baseline models are enhanced with the traditional methods of imbalance handling, namely SMOTE and adaptive resampling.
- Integrated: With GAN augmentation (in particular, on-the-fly GAN augmentation of the imbalance detection to create synthetic minority samples, e.g., Botnet) injected into retraining as and when the imbalance is detected.
- Traffic is played through tcpreplay at different rates (1 K-10K packets/s) with a split ratio of 70/15/15 for training/validation/testing. All the experimental settings were performed on 10 independent runs to assure stringent statistical authentication. The findings on the measures of minority-class recall and F1-score contain the variance reporting to confirm the consistency and reliability of the proposed event-driven augmentation against the ordinary background variation.

6. Discussion and Potential Extensions

The suggested GAN-coupled IDS system has strong points to improve the work of cybersecurity units, such as the option to adjust in time to intrusion detection and the ability to overcome the delays of the dataset (CIC-IDS-2017 and CIC-IDS-2018) caused by class imbalance. The system makes it possible to generate synthetic data on-the-fly to achieve dynamic model retraining, a crucial factor in increasing the detection of rare attacks such as U2R and R2L without necessitating large samples in the real world, as demonstrated by earlier tests with up to 98.2% accuracy on augmented random forest classifiers. This flexibility works with changing threat landscapes, which means that it can work in an operational environment where conventional IDS have problems with asymmetrical traffic patterns.

Nevertheless, various issues should be considered, such as ethical and security-related risks of using synthesized information in the wrong way, including opponents producing fake datasets that are used to maintain discrimination or even data poisoning assault in cybersecurity applications. Scalability is another issue, since GAN training and real-time integration may require a lot of computer power; therefore, in a large network, there are chances of latency or an overall decline in efficiency in high-traffic environments.

Future work may involve the addition of feedback loops with IDS-detected anomalies acting to train the GAN so that it produces the desired specialized fake samples to form an intelligent loop that constructs a sub-awareness able to detect certain patterns, such as a type of DDoS. The final extension is the edge or distributed deployment, in which GAN components are deployed on the edges of the network in support of federated learning to maintain privacy by reducing the amount of data centralization but allowing mutual model updates within any IoT or cloud-edge infrastructure. These extensions are very sensitive from a security perspective, and a strong validation system, such as a check on differential privacy or

adversarial robustness, would not allow synthetic data to provide susceptibilities to the data, such as re-identification possibilities or biased evaluations.

It is desirable to conduct full-scale deployment trials within practical networks in the future to ensure the feasibility of the architecture in real life beyond simulations, as well as to consider conditional GANs in generating labeled sets of samples of attacks to potentially augment targeted closer to particular forms of intrusion.

7. Conclusion

This study introduces a feasible framework for the integration of GAN-based synthetic data into a live IDS framework, which closes the gap between offline augmentation and operational implementation by utilizing a pipeline encompassing traffic observation, anomaly-based generation, and marginal retraining. Its feasibility was achieved through a prototype with Suricata and the CIC-IDS-2017/2018 dataset, with all projected advantages shown (e.g., successful gains in minority class identification, e.g., 3-5% accuracy increase) and low latency supported (e.g., quantitative measurements) and assessed (e.g., qualitative bar chart analysis).

The effect on cybersecurity is significant since such a procedure proves to be a sure avenue towards more robust operational IDS to deal with dynamic threats, such as adaptive attacks in the scope of IoT and cloud networks, due to the continuous enhancement of data that balances the imbalance in classification and prevents the emergence of false positives. We propose additional research in online augmentation tools like later iterations, like conditional GANs and federated learning integration, to scale such solutions to larger publicly utilized cybersecurity practices currently.

References

[1] J. Li, W. Zong, Y.-W. Chow, and W. Susilo, "Mitigating Class Imbalance in Network Intrusion Detection with Feature-Regularized GANs," *Future Internet*, vol. 17, no. 5, p. 216, 2025, doi: 10.3390/fi17050216.

[2] M. Al-Ajlan and M. Ykhlef, "A Review of Generative Adversarial Networks for Intrusion Detection Systems: Advances, Challenges, and Future Directions," *Comput. Mater. Contin.*, vol. 81, no. 2, pp. 2053–2076, 2024, doi: 10.32604/cmc.2024.055891.

[3] M. A. Rahman, G. A. Francia, and H. Shahriar, "Leveraging gans for synthetic data generation to improve intrusion detection systems," *J. Future Artif. Intell. Technol.*, vol. 1, no. 4, pp. 429–439, 2025, doi: 10.62411/faith.3048-3719-52.

[4] S. Menssouri and E. M. Amhoud, "A Conditional Tabular GAN-Enhanced Intrusion Detection System for Rare Attacks in IoT Networks," Feb. 09, 2025, *arXiv*. doi: 10.48550/arXiv.2502.06031.

[5] Y. Yang *et al.*, "A CE-GAN based approach to address data imbalance in network intrusion detection systems," *Sci Rep*, vol. 15, p. 7916, 2025, doi: 10.1038/s41598-025-90815-5.

[6] C. E. L. Asry, I. Benchaji, S. Douzi, and B. E. L. Ouahidi, "Enhancing cybersecurity: A high-performance intrusion detection approach through boosting minority class recognition," *PLOS One*, vol. 20, no. 3, p. e0317346, Mar. 2025, doi: 10.1371/journal.pone.0317346.

[7] P. Yadav, G. Sihag, and V. Vijay, "Rebalancing the Scales: A Systematic Mapping Study of Generative Adversarial Networks (GANs) in Addressing Data Imbalance," Feb. 23, 2025, *arXiv*: arXiv:2502.16535. doi: 10.48550/arXiv.2502.16535.

[8] G. Agrawal, A. Kaur, and S. Myneni, "A review of generative models in generating synthetic attack data for cybersecurity," *Electronics*, vol. 13, no. 2, p. 322, 2024, doi: 10.3390/electronics13020322.

[9] J. Lee, D. Jung, J. Moon, and S. Rho, "Advanced R-GAN: Generating anomaly data for improved detection in imbalanced datasets using regularized generative adversarial networks," *Alex. Eng. J.*, vol. 111, pp. 491–510, Jan. 2025, doi: 10.1016/j.aej.2024.10.084.

[10] "Class imbalanced data handling with cyberattack classification using Hybrid Salp Swarm Algorithm with deep learning approach," *Alex. Eng. J.*, vol. 106, pp. 654–663, Nov. 2024, doi: 10.1016/j.aej.2024.08.061.

[11] A. G. Udu, M. T. Salman, M. K. Ghalati, A. Lecchini-Visintini, D. R. Siddle, and H. Dong, "Emerging SMOTE and GAN Variants for Data Augmentation in Imbalance Machine Learning Tasks: A Review," *IEEE Access*, vol. 13, pp. 113838–113853, 2025, doi: 10.1109/ACCESS.2025.3584532.

[12] R. Sauber-Cole and T. M. Khoshgoftaar, "The use of generative adversarial networks to alleviate class imbalance in tabular data: a survey," *J. Big Data*, vol. 9, no. 1, p. 98, Aug. 2022, doi: 10.1186/s40537-022-00648-6.

[13] E. Strelcenia and S. Prakoonwit, "A Survey on GAN Techniques for Data Augmentation to Address the Imbalanced Data Issues in Credit Card Fraud Detection," *Mach. Learn. Knowl. Extr.*, vol. 5, no. 1, Art. no. 1, Mar. 2023, doi: 10.3390/make5010019.

[14] X. Zhao, K. W. Fok, and V. L. Thing, "Enhancing network intrusion detection performance using generative adversarial networks," *Comput. Secur.*, vol. 145, p. 104005, 2024.

[15] "Deep learning-driven methods for network-based intrusion detection systems: A systematic review," *ICT Express*, vol. 11, no. 1, pp. 181–215, Feb. 2025, doi: 10.1016/j.icte.2025.01.005.

[16] W. Almuseelem, "Perspective Chapter: Intrusion Detection Systems in Cloud Environment," *Mastering Intrusion Detect. Cybersecurity*, 2025, doi: 10.5772/intechopen.1008756.
[17] O. E. Aeraj and C. Leghris, "Study of the SNORT intrusion detection system based on machine learning," in *2023 7th IEEE Congress on Information Science and Technology (CiSt)*, Dec. 2023, pp. 33–37. doi: 10.1109/CiSt56084.2023.10409876.
[18] A. Oun, K. Wince, and X. Cheng, "The Role of Artificial Intelligence in Boosting Cybersecurity and Trusted Embedded Systems Performance: A Systematic Review on Current and Future Trends," *IEEE Access*, vol. 13, pp. 55258–55276, 2025, doi: 10.1109/ACCESS.2025.3554739.
[19] W. Lim, K. S. C. Yong, B. T. Lau, and C. C. L. Tan, "Future of generative adversarial networks (GAN) for anomaly detection in network security: A review," *Elsevier*, vol. 139, p. 103733, 2024, doi: 10.1016/j.cose.2024.103733.
[20] "An intelligent intrusion detection system for cyber-physical systems using GAN-LSTM networks," *Frankl. Open*, vol. 11, p. 100281, Jun. 2025, doi: 10.1016/j.fraope.2025.100281.
[21] "SYN-GAN: A robust intrusion detection system using GAN-based synthetic data for IoT security," *Internet Things*, vol. 26, p. 101212, Jul. 2024, doi: 10.1016/j.iot.2024.101212.
[22] M. Poongodi and M. Hamdi, "Intrusion detection system using distributed multilevel discriminator in GAN for IoT system," *Trans. Emerg. Telecommun. Technol.*, vol. 34, no. 11, doi: doi.org/10.1002/ett.4815.
[23] D. Singh, B. Raman, A. K. Luhach, and P. Lingras, Eds., *Advanced Informatics for Computing Research*. in Communications in Computer and Information Science. Singapore: Springer, 2017. doi: 10.1007/978-981-10-5780-9.
[24] H. N. Nguyen, T. Lan-Phan, and C.-J. Song, "Generative Adversarial Network-Based Network Intrusion Detection System for Supervisory Control and Data Acquisition System," in *2024 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia)*, Nov. 2024, pp. 1–3. doi: 10.1109/ICCE-Asia63397.2024.10773791.
[25] "Generative adversarial networks-enabled anomaly detection systems: A survey," *Expert Syst. Appl.*, vol. 296, p. 128978, Jan. 2026, doi: 10.1016/j.eswa.2025.128978.
[26] V. Kumar and D. Sinha, "Synthetic attack data generation model applying generative adversarial network for intrusion detection," *Comput. Secur.*, vol. 125, p. 103054, 2023, doi: 10.1016/j.cose.2022.103054.
[27] S. K. Erskine, "Real-Time Large-Scale Intrusion Detection and Prevention System (IDPS) CICIoT Dataset Traffic Assessment Based on Deep Learning," *Appl Syst Innov*, vol. 8, no. 2, 2025, doi: 10.3390/asi8020052.
[28] S. Kaushik *et al.*, "Robust machine learning based Intrusion detection system using simple statistical techniques in feature selection," *Sci. Rep.*, vol. 15, no. 1, p. 3970, Feb. 2025, doi: 10.1038/s41598-025-88286-9.
[29] S. Aldhaheri and A. Alhuzali, "SGAN-IDS: Self-Attention-Based Generative Adversarial Network against Intrusion Detection Systems," *Sensors*, vol. 23, no. 18, Art. no. 18, Jan. 2023, doi: 10.3390/s23187796.
[30] D. Xu and B. Cao, "Adaptive Multiobjective Evolutionary Generative Adversarial Network for Metaverse Network Intrusion Detection," *Research*, vol. 8, p. 0665, doi: 10.34133/research.0665.
[31] M. Ozkan-Okay *et al.*, "A Comprehensive Survey: Evaluating the Efficiency of Artificial Intelligence and Machine Learning Techniques on Cyber Security Solutions," *IEEE Access*, vol. 12, pp. 12229–12256, 2024, doi: 10.1109/ACCESS.2024.3355547.
[32] P. V. Chavan and N. V. Alone, "Optimizing Intrusion Detection with Random Forest: A High-Accuracy Approach using CIC-IDS 2017," *Int. J. Comput. Appl.*, vol. 187, no. 3, pp. 17–22.
[33] M. M. Rahma, Shaharia Al Shakil, and M. R. Mustakim, "A survey on intrusion detection system in IoT networks," *Cyber Secur. Appl.*, vol. 3, no. 100082, Dec. 2025, doi: 10.1016/j.csa.2024.100082.