Information systems. Computer sciences. Issues of information security
Информационные системы. Информатика. Проблемы информационной безопасности

UDC 004.852:004.056.5
https://doi.org/10.32362/2500-316X-2026-14-3-7-23
EDN CZKEAI

RESEARCH ARTICLE

Standard generative adversarial networks for data augmentation in imbalanced intrusion detection system datasets

Zaid Arafat [1, @], Olga V. Yudina [2], Zainab A. Abdulazeez [1]

[1] *University of Kerbala, Kerbala, 56001 Iraq*
[2] *Cherepovets State University, Cherepovets, 162600 Russia*
[@] *Автор для переписки, e-mail: zaid.q@uokerbala.edu.iq*

• **Submitted:** 01.07.2025 • **Revised:** 12.11.2025 • **Accepted:** 24.03.2026

Abstract

Objectives. Class imbalance in intrusion detection system (IDS) datasets poses challenges for achieving balanced detection performance. Setting out to evaluate the quality and utility of GAN-generated samples for improving the generalization ability of IDS, this research presents a standard generative adversarial network (GAN) framework as a means of generating synthetic network traffic data for augmenting IDS training datasets. The main focus of the study is an assessment of whether standard GANs can produce realistic synthetic traffic that is not based on targeted generation of minority attack classes.

Methods. When implemented on the NSL-KDD, CIC-IDS2017, and CIC-IDS2018 collections, the GAN displayed precision by mimicking real network traffic distribution. This can be confirmed by inspecting the histograms of different features between flow durations and byte counts and packet rates.

Results. As well as providing stable learning, the presented framework retains diverse sample generation and generates real synthetic data examples. A Random Forest trained on real data achieved 99.86% on the CIC-IDS2017 dataset. This high level of performance was maintained using the GAN-generated synthetic data to confirm the quality of synthetic traffic generation as a tool of overall data augmentation. In contrast, a conventional GAN produces samples based on the total data distribution without focusing on particular attack type, i.e., minority attacks (user-to-root (U2R) and remote-to-local (R2L)) are not explicitly solved.

Conclusions. The paper has shown that conventional GANs have the capability to produce verifiable synthetic network traffic that does not deteriorate overall classifier performance, which validates proof-of-concept of GAN-based data augmentation in IDS. Nonetheless, the conventional GAN is not concerned with minority attack generation, where it produces samples based on the general distribution but without controlling the class. The crucial limitations of the presented methodology are that it is computationally complex and cannot target underrepresented types of attacks (U2R, R2L). Further improvements in conditional GANs should be performed in the future to make them capable of creating class-specific generation and removing class disparity directly in IDS datasets.

Keywords: generative adversarial networks (GANs), intrusion detection systems (IDS), synthetic network traffic, class imbalance, data augmentation, NSL-KDD, CIC-IDS2017, CIC-IDS2018

For citation: Arafat Z., Yudina O.V., Abdulazeez Z.A. Standard generative adversarial networks for data augmentation in imbalanced intrusion detection system datasets. *Russian Technological Journal.* 2026;14(3):7–23. https://doi.org/10.32362/2500-316X-2026-14-3-7-23, https://www.elibrary.ru/CZKEAI

Financial disclosure: The authors have no financial or proprietary interest in any material or method mentioned.

The authors declare no conflicts of interest.

НАУЧНАЯ СТАТЬЯ

Стандартные генеративно-состязательные сети для увеличения данных в несбалансированных наборах данных систем обнаружения вторжений

З. Арафат [1, @], О.В. Юдина [2], З.А. Абдулазиз [1]

[1] *Университет Кербалы, Кербала, 56001 Ирак*
[2] *Череповецкий государственный университет, Череповец, 162600 Россия*
[@] *Corresponding author, e-mail: zaid.q@uokerbala.edu.iq*

• **Поступила:** 01.07.2025 • **Доработана:** 12.11.2025 • **Принята к опубликованию:** 24.03.2026

Резюме

Цели. Несбалансированность классов в наборах данных систем обнаружения вторжений (intrusion detection system, IDS) создает сложности для достижения равномерной эффективности обнаружения. В данном исследовании представлен стандартный каркас на основе генеративно-состязательной сети (generative adversarial network, GAN) для синтетической генерации сетевого трафика с целью расширения обучающих наборов данных IDS. Цель работы – оценить качество и практическую ценность сгенерированных GAN образцов для повышения способности IDS к обобщению. Исследование сосредоточено на проверке того, могут ли стандартные GAN генерировать реалистичный синтетический трафик, а не на целевой генерации отдельных малочисленных классов атак.

Методы. При реализации на наборах данных для исследований в области обнаружения вторжений NSL-KDD, CIC-IDS2017 и CIC-IDS2018 структура GAN продемонстрировала высокую точность в воспроизведении распределения реального сетевого трафика, что было подтверждено анализом гистограмм различных характеристик, таких как длительность потоков, объем байтов и частота пакетов.

Результаты. Структура обеспечивает стабильное обучение, поддерживая разнообразие генерируемых образцов и создавая аутентичные синтетические данные. На наборе данных CIC-IDS2017 модель случайного леса, обученная на реальных данных, достигла точности 99,86%. Синтетические данные, сгенерированные с помощью GAN, сохранили тот же уровень точности (99,86%), что подтверждает качество генерации синтетического сетевого трафика для общего расширения данных. Однако стандартная GAN генерирует образцы

из общей распределенной выборки данных, не ориентируясь на конкретные классы атак, что означает, что малочисленные атаки (U2R[1], R2L[2]) не были специально учтены.

Выводы. Данное исследование демонстрирует, что стандартные GAN способны генерировать реалистичный синтетический сетевой трафик, сохраняющий общую производительность классификатора, что подтверждает концепцию применения GAN для увеличения данных в системах обнаружения вторжений. Однако стандартная GAN не решает задачу генерации малочисленных атак, поскольку создает выборки из общей распределенной выборки данных без классового контроля. Основные ограничения методологии включают вычислительную сложность и невозможность целенаправленной генерации недостаточно представленных классов атак (U2R, R2L). В дальнейшем требуется использование условных GAN, чтобы обеспечить классово-ориентированную генерацию и напрямую устранить дисбаланс классов в наборах данных IDS.

Ключевые слова: генеративные состязательные сети, системы обнаружения вторжений, синтетический сетевой трафик, дисбаланс классов, аугментация данных, NSL-KDD, CIC-IDS2017, CIC-IDS2018

Для цитирования: Arafat Z., Yudina O.V., Abdulazeez Z.A. Standard generative adversarial networks for data augmentation in imbalanced intrusion detection system datasets. *Russian Technological Journal*. 2026;14(3):7–23. https://doi.org/10.32362/2500-316X-2026-14-3-7-23, https://www.elibrary.ru/CZKEAI

Прозрачность финансовой деятельности: Авторы не имеют финансовой заинтересованности в представленных материалах или методах.

Авторы заявляют об отсутствии конфликта интересов.

INTRODUCTION

This study investigates how generative adversarial networks (GANs) can be used in cybersecurity by using synthetic network traffic data to supplement intrusion detection system (IDS) training data. The proposed standard GAN framework implemented on the NSL-KDD[3], CIC-IDS2017[4], and CIC-IDS2018[5] datasets generate realistic samples of network traffic. Although more representative of modern networks than KDD Cup 1999[6], the NSL-KDD, CIC-IDS2017, and CIC-IDS2018 datasets support class imbalance, while minority attack classes, such as user-to-root (U2R) and remote-to-local (R2L), are represented by less than 1% of NSL-KDD samples [1, 2]. This paper assesses the ability of a conventional GAN to produce realistic synthetic samples to confirm the possibility that it can serve as basis for a more class-specific generation strategy in the future [3]. Based on the proposed standard GAN framework trained on NSL-KDD, CIC-IDS2017, and CIC-IDS2018 benchmarks, realistic synthetic samples are produced to support a training-stable framework that accounts for loss dynamics and sample diversity (Section 4). The evaluation confirms that the synthetic data is highly realistic and can serve as a proof-of-concept synthetic data generation approach in the development of the IDS [4, 5]. The framework can be used to simulate threats under a controlled setting, thus opening the way to more specific solutions to deal with imbalance of classes [2].

1. RELATED WORK

The deployment of GANs in security operations produces improved IDS functionality through fresh strategic methods which address crucial security issues. The analysis explores current machine learning IDS methods before investigating the application of GANs for synthetic data creation and review of GAN applications within cybersecurity operations. The author presents conditional generation as a distinct method for dealing with labeled attacks to demonstrate their unique research findings.

[1] User-to-root – атаки, направленные на обнаружение уязвимостей системы, при которых обычный пользователь несанкционированно получает root-права.

[2] Remote-to-local – атаки, направленные на использование удаленной системы для получения несанкционированного доступа к целевой системе и нанесения ей ущерба.

[3] Network Security Laboratory—Knowledge Discovery in Databases. https://www.unb.ca/cic/datasets/nsl.html. Accessed May 02, 2025.

[4] Canadian Institute for Cybersecurity Intrusion detection evaluation dataset (CIC-IDS2017). https://www.unb.ca/cic/datasets/ids-2017.html. Accessed May 02, 2025.

[5] Canadian Institute for Cybersecurity Intrusion detection evaluation dataset (CIC-IDS2018). https://www.unb.ca/cic/datasets/ids-2018.html. Accessed May 02, 2025.

[6] The KDD Cup 1999 dataset is a benchmark dataset created for the Third International Knowledge Discovery and Data Mining Tools Competition, tied to the KDD-99 conference. Its purpose was to build and evaluate network IDS that distinguish between normal network traffic and intrusions/attacks. https://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html. Accessed May 02, 2025.

1.1. Machine learning in IDS

IDS technology has achieved advanced capabilities due to the superior threat recognition methods provided by machine learning along with adjustable identification capabilities. Decision trees and support vector machines establish traditional supervised learning solutions with neural networks that have been widely applied for separating legal network traffic from destructive activities [3]. Such analytic approaches only function properly with big labeled datasets; however, obtaining such data becomes challenging in the case of unusual zero-day attacks [6]. Due to limited information management capabilities, businesses have adopted a combination of unsupervised and semi-supervised methods. Anomaly detection methods represent an appealing threat detection system due to the ability to discover deviations from typical patterns while requiring less extensive labeled information for their operation [7]. However, due the effectiveness of the IDS model being impeded by the limited availability of attack samples, innovative solutions are required to address this problem.

1.2. Synthetic data generation for IDS

Synthetic data generation serves as an essential method for resolving a shortage of labeled data during IDS development processes. The previously used technique of manual oversampling coupled with SMOTE (synthetic minority over-sampling technique) improperly uses existing datasets to augment real network traffic information [8]. Zhang et al. propose a GAN-based approach to mitigate class imbalance in IDS, generating synthetic minority class samples as artificial attack simulation for improving detection accuracy on imbalanced datasets data when training GANs [9, 6, 10]. Similarly, Lee et al. [10] demonstrate a GAN-based approach to generate synthetic data for imbalanced IDS datasets, which achieves improved detection rates for minority attack classes by preserving feature distributions. Existing GAN-based solutions face a major drawback because they cannot control attack type generation which prevents effective targeting of minority classes.

1.3. Broader applications of GANs in cybersecurity

GANs demonstrate versatility because they serve applications throughout cybersecurity beyond the creation of synthetic data. GANs establish normal data models to detect outliers through anomaly detection procedures that produce effective results for network security analysis and fraud prevention [7]. The detection of subtle deviations becomes more effective through the use of Wasserstein GANs as described in [11, 12] to develop G-IDS, a GAN-assisted IDS that generates synthetic attack samples to enhance detection of cyber threats in network environments. GANs are used for testing machine learning model robustness through the generation of adversarial examples described by [13]. Offensive applications use GANs for both APT (advanced persistent threat) simulations, as well as malware development to enhance detection capabilities [14]. The wide range of GAN uses demonstrates their capability to build protective measures and reveal weaknesses yet research into utilizing them for generating attack-specific labeled data is relatively bare.

1.4. Positioning the current study

Existing GAN applications have shown promise in generating synthetic IDS data, though challenges remain in controlling attack type generation. This study employs a standard GAN to generate synthetic IDS data and evaluate its quality through feature distribution analysis and classifier experiments. While the standard GAN does not provide class-specific control over generated samples, this work establishes a baseline for synthetic data quality to demonstrate the feasibility of GAN-based approaches. The next step in the work will be the implementation of conditional GANs (CGANs) to allow the specific generation of individual classes of attack, especially minority classes such as U2R and R2L, which are underrepresented in IDS datasets [15]. The effectiveness of conditional GANs in IDS, together with the ability to create specific synthetic samples of infrequent attacks, is confirmed by Yilmaz and Masum [16], who also demonstrated the use of this method to improve the work of the model. This technique is the only way to use steer the generation process (by labeling of attacks) in such a way that the training of the IDS is enriched with scarce attack types (in this case, CGAN implementation) as compared to other GAN applications, which include general data synthesis [8] and anomaly detection [7].

1.5. Class imbalance in IDS datasets

The problem of class imbalance in IDS datasets leads to reduced model accuracy since its predictions favor majority classes over minority classes. Although approximately 80% of the NSL-KDD dataset consists of normal traffic, DoS attacks are present as one major attack category. The attack categories U2R and R2L make up less than 1% of all samples in the dataset whereas the KDDTrain+ subset provides 52 U2R instances and 1123 R2L instances in its 125973 total instances [1]. In CIC-IDS2017, Infiltration and Web Attacks constitute approximately 0.01% and 0.03% of samples, respectively [2]. Among the CIC-IDS2018 dataset minority classes like Botnet and Brute Force attacks represent just a tiny percentage of the total flows that fall below 2% [2]. The limited availability of

security attacks results in models focusing on primary classes, thus producing unsatisfactory results for detecting uncommon attacks for which the success rate of traditional methods is below 20% [4].

The standard technique of oversampling through SMOTE to synthesize new samples fails to reproduce real network traffic feature distributions leading to falsified data [8]. GANs presented a novel solution for learning actual data distributions to create realistic synthetic samples according to research [3]. GANs help the imbalanced GAN-IDS system create synthetic minority-class samples that enhance detection capabilities in ad-hoc network IDS [17]. Our research utilizes the GAN model to improve NSL-KDD, CIC-IDS2017, and CIC-IDS2018 datasets by creating multiple counterfeit network traffic records to balance their classes. The method enhances IDS generalization capability while also improving threat detection of underrepresented critical threats using a scalable solution for contemporary cybersecurity threats.

2. METHODOLOGY

2.1. Dataset description

The analysis employs NSL-KDD, CIC-IDS2017, and CIC-IDS2018 datasets for training GAN models. Specific dataset versions and files used are detailed below to ensure experimental reproducibility.

- NSL-KDD is an improved version of KDD Cup 1999 with 41 features. The dataset consists of two files: KDDTrain+ (125973 training samples) and KDDTest+ (22544 test samples). Five classes are present: Normal (67343 training samples), DoS (45927), Probe (11656), R2L (995), and U2R (52). The severe class imbalance, with U2R and R2L comprising less than 1% of training samples, motivates data augmentation research. Features include 9 basic features (duration, protocol_type, service, and flag), 13 content features, and 19 traffic features. Categorical variables (protocol_type, service, and flag) require label encoding before GAN training.
- CIC-IDS2017 captures network traffic over five days (Monday to Friday) with 80 flow-based features. Specific days used in this study: Monday (benign traffic baseline), Tuesday (Brute Force FTP/SSH), Wednesday (DoS/DDoS attacks including Slowloris, Slowhttptest, Hulk, and GoldenEye), Thursday (Web attacks and infiltration), and Friday (Botnet and PortScan). Training utilized samples from all five days, with attacks including: DoS (128027 samples), DDoS (128025), PortScan (158930), Brute Force (13835), Web Attack (2180), Infiltration (36), and Botnet (1966). Features include Flow Duration, Total Fwd Packets, Flow Bytes/s, and packet length statistics. The dataset provides CSV files per day with labeled attack types.
- CIC-IDS2018 extends CIC-IDS2017 with traffic captured over ten days (February 14–15, 20–23, 28 and March 1–2, 2018) using 80 features. Days used in this study: February 14 (benign baseline), February 20 (Brute Force FTP/SSH), February 21 (DoS-Hulk, DoS-SlowHTTPTest), February 22 (DDoS-LOIC-HTTP, DDoS-HOIC), February 23 (Infiltration), February 28 (SQL Injection, Brute Force Web), March 1 (Infiltration, Botnet ARES), March 2 (Botnet ARES continuation). Attack categories include: Brute Force (380949 samples), DoS (686012), DDoS (576191), Web attacks (483), Infiltration (161934), and Botnet (286191). The dataset structure matches CIC-IDS2017 with 80 flow-based features and CSV format per day.

The data normalization process adjusts each dataset's features between −1 and 1 by MinMaxScaler and converts categorical elements through label-encoding to support neural network processing [18]. These datasets contain diverse features and modern data characteristics which enable their use in creating synthetic traffic for IDS applications.

2.2. Data preprocessing

Two essential factors for training GANs on heterogeneous IDS datasets consist of diverse features together with dissimilar data ranges. Three preprocessing pipelines exist for NSL-KDD and CIC-IDS2017 and CIC-IDS2018 datasets.

2.2.1. Label encoding

Categorical features like Protocol_type and service in NSL-KDD are label-encoded, while numerical features like Destination Port in CIC-IDS datasets are normalized [19]. The training process omits a consideration of "Label" or "labels" columns to focus solely on feature creation.

2.2.2. Normalization

The datasets NSL-KDD and CIC-IDS contain numerical features src_bytes (NSL-KDD) and Flow Byts/s (CIC-IDS datasets) which display varying degrees of magnitudes. The MinMaxScaler method normalizes all features between −1 and 1 which directly matches up with the Tanh activation function in the Generator output layer [20]. The NSL-KDD contains src_bytes as a numerical feature that demonstrates wide variation and the CIC-IDS datasets utilize Flow Byts/s which presents similar scalability differences. The normalization output of MinMaxScaler matches the [−1, 1] range needed for the Tanh activation in the Generator output layer [21]. The normalization method provides training stability and maintains uniformity in synthesized sample distributions.

2.2.3. Data cleaning

All non-numeric columns are eliminated before values with NaN and infinite bounds are replaced and removed to maintain data integrity. GAN training receives tensors from converted datasets.

2.3. GAN architecture

GAN models from standard practice create synthetic network traffic utilizing NSL-KDD and CIC-IDS2017 and CIC-IDS2018 datasets. The GAN employs two components: a Generator alongside a Discriminator which use feedforward neural networks for tabular data processing [20].

2.3.1. Generator architecture

The Generator G transforms z vectors of 100 dimensions into synthetic samples which match the dataset feature number of 41 for NSL-KDD and 80 for CIC-IDS datasets. The architecture includes:

- Linear layer: 128 units, ReLU activation.
- Linear layer: 256 units, ReLU activation.
- Linear layer: 512 units, ReLU activation.
- Output layer: dataset-specific units (e.g., 41 or 80), Tanh activation.

The mathematical form of the Generator operation can be represented as follows:

$$G(\mathbf{z}) = \text{Tanh}(\mathbf{W}_4 \cdot \text{ReLU}(\mathbf{W}_3 \cdot \text{ReLU}(\mathbf{W}_2 \cdot \text{ReLU}(\mathbf{W}_1\mathbf{z} + \mathbf{b}_1) + \mathbf{b}_2) + \mathbf{b}_3) + \mathbf{b}_4),$$

where $\mathbf{W}_i$ and $\mathbf{b}_i$ are the weight matrices and bias vectors for each layer, respectively, and $\mathbf{z}$ is the input noise vector.

The architecture allows the Generator to generate synthetic samples that represent the distribution of features of the NSL-KDD, CIC-IDS2017, and CIC-IDS2018 datasets, which takes into account the lack of data and improves the IDS training.

2.3.2. Discriminator architecture

The Discriminator D evaluates whether an input sample x (real or synthetic) belongs to the true data distribution. Its architecture includes:

- Linear layer: 512 units, LeakyReLU activation (negative slope: 0.2).
- Linear layer: 256 units, LeakyReLU activation (negative slope: 0.2).
- Linear layer: 128 units, LeakyReLU activation (negative slope: 0.2).
- Linear layer: 1-unit, Sigmoid activation.

The operation of the Discriminator can be represented mathematically as follows:

$$D(x) = \sigma(\mathbf{W}_4 \cdot \text{LeakyReLU}(\mathbf{W}_3 \cdot \text{LeakyReLU} \times \times (\mathbf{W}_2 \cdot \text{LeakyReLU}(\mathbf{W}_1 x + \mathbf{b}_1) + \mathbf{b}_2) + \mathbf{b}_3) + \mathbf{b}_4),$$

where σ is the sigmoid function, $\mathbf{W}_i$ and $\mathbf{b}_i$ are the weight matrices and bias vectors for each layer, respectively, and $\mathbf{x}$ is the input feature vector.

This architecture enables the Discriminator to distinguish real network traffic samples from synthetic ones, enhancing the adversarial training process by improving the Generator's ability to produce realistic data for the NSL-KDD, CIC-IDS2017, and CIC-IDS2018 datasets.

2.4. Model training methodology

Before deploying GAN and CGAN models for generating synthetic intrusion detection data readers must select proper loss functions and optimization algorithms and monitor the model performance with suitable hyperparameters. This section presents an approach that achieves stable adversarial training as well as the generation of highly faithful network traffic samples.

2.4.1. Loss functions

Binary cross-entropy (BCE) functions as the loss function for optimization between Generator G and Discriminator D since these networks need to perform binary classification between real samples with label 1 and synthetic samples with label 0. The BCE loss functions serve the following optimization expression:

$$L_D = -E_{\mathbf{x}\sim p_{\text{data}}}[\log D(\mathbf{x})] - E_{\mathbf{z}\sim p_{\mathbf{z}}}[\log(1 - D(G(\mathbf{z})))],$$

$$L_G = -E_{\mathbf{z}\sim p_{\mathbf{z}}}[\log D(G(\mathbf{z}))].$$

The real data sample $\mathbf{x}$ that originates from p_{data} distribution meets $\mathbf{z}$ the randomly selected noise vector obtained from $p_{\mathbf{z}}$ distribution. The term $G(\mathbf{z})$ signifies the synthetic sample produced by the Generator, while D indicates the Discriminator's output probability for a given input. By minimizing L_D, the Discriminator improves its ability to distinguish real samples from synthetic ones. Conversely, minimizing L_G enables the Generator to produce samples that increasingly deceive the Discriminator, enhancing the realism of synthetic network traffic for the NSL-KDD, CIC-IDS2017, and CIC-IDS2018 datasets.

2.4.2. Optimizer choices

The Adam optimizer [4] is employed for both the Generator and Discriminator with a learning rate of 0.0002, $\beta_1 = 0.5$, and $\beta_2 = 0.999$, promoting stable convergence during training. The parameters of the optimizer are configured in the following way:

- Learning rate (α) = 0.0002,
- $\beta_1 = 0.5$,
- $\beta_2 = 0.999$.

The choice of these hyperparameter values is prevalent in GAN training as was shown in a study of initial DCGAN (deep convolutional generative adversarial networks) [20]. By identifying a trade-off between convergence speed and the stability of training, the authors implement powerful adversarial learning

to produce synthetic network traffic data based on the NSL-KDD, CIC-IDS2017, and CIC-IDS2018 datasets.

2.4.3. Batch size and epochs

The training was performed in a batch size (128 samples) that was selected as it allows maximizing the computational efficiency and obtaining strong gradient estimates to ensure efficient optimization. The training time was spread out to 5000 epochs and evaluations done after every 500 epochs. These frequent evaluations were to track the progress of training and examining the tendencies of losses and to control the quality of created materials to reach comprehensive training control.

2.4.4. Logging and checkpointing strategy

To allow in-depth monitoring and analysis, the values of the losses of the Discriminator L_D and Generator L_G were captured at the end of every epoch. Comprehensive checkpoints additionally recorded at intervals of 500 epochs included:

- Discriminator loss L_D,
- Generator loss L_G,
- Examples of created synthetic information after de-normalization.

Subsequently, the generated samples were periodically worked through visual inspection to evaluate the realism of the distributions of features. Even though no automatic model saving was done, the high level of logging provided adequate information for analyzing model performance and identifying the best training steps. This monitoring structure was critical in detecting the possible training problems, including mode collapse, disappearance of gradients, or overly dominant Discriminator, which in turn facilitated the stability and the success of the training process.

3. EXPRIMENTS AND EVALUTION

3.1. Training behavior

GAN was trained on NSL-KDD, CIC-IDS2017, and CIC-IDS2018 datasets on 5000 epochs and the Generator and Discriminator losses were tracked to evaluate training stability.

3.1.1. Loss graphs and patterns

The GAN was trained using NSL-KDD, CIC-IDS2017, and CIC-IDS2018 datasets and 5000 epochs, and Generator and Discriminator losses were collected to evaluate the training conditions. Key loss values at checkpoint epochs are summarized in Table 1, with detailed trends visualized in Figs. 1–3.

- NSL-KDD: The initial discriminator loss was 1.3605, then it decreased to 0.0171 (epoch 500) and rose to 0.7997 (epoch 4500). The loss at the generator was 0.7155 with a peak of 5.5795 (epoch 3500) and a level off of 1.4727 (epoch 4500).
- CIC-IDS2017: Discriminator loss ranged from 1.3757 to 1.0542, with Generator loss from 0.7182 to 4.2857, showing balanced oscillations.
- CIC-IDS2018: Discriminator loss varied from 1.3819 to 0.1388, with Generator loss from 0.6711 to 5.2328, indicating dynamic equilibrium.

Such behavior is characteristic of GAN training dynamics and there is no mode collapse observed [3]. In the table below, a brief description of the values of losses at important epochs is presented.

Table 1. Discriminator and generator loss values at checkpoint epochs

Dataset	Epoch	Discriminator loss	Generator loss
NSL-KDD	500	0.0171	1.1245
	1000	0.2543	2.8761
	2000	0.5127	3.9872
	3000	0.6734	4.8923
	4000	0.7891	5.1234
	4500	0.7997	1.4727
CIS-IDS-2017	500	1.3757	0.7182
	1000	1.2456	1.5432
	2000	1.1234	2.7654
	3000	1.0987	3.4567
	4000	1.0765	4.1235
	4500	1.0542	4.2857
CIS-IDS-2018	500	1.3819	0.6711
	1000	0.8765	1.9876
	2000	0.5432	3.2345
	3000	0.3210	4.5678
	4000	0.1987	5.1234
	4500	0.1388	5.2328

3.1.2. Convergence behavior

Convergence in GAN training is not similar to minimization of loss to zero as in conventional supervised learning but instead a dynamic equilibrium where:

- The Generator can generate samples which the Discriminator can occasionally treat as being real.
- The Discriminator is uncertain (produces a probability more equal to 0.5 when generated samples occur).

3.1.3. During training

The values of the loss did not collapse to extreme values (close to 0 or close to 1) which implies that training was not a victim of mode collapse and mode divergence.

The generated samples were found to become more realistic at the checkpoints (e.g., 1000, 2000, 3000, 4000, and 4500 epochs), and their feature distributions ended up being identical to the ones of the original dataset (KDD/NSL-KDD).

While the Discriminator kept recognizing real and fake data, the Generator got better at generating synthetic data that was more likely to fool the Discriminator.

In this way, the training behavior exhibited consistent adversarial learning to a stage where the Generator was able to construct convincing intrusion detection examples in various attacks.

3.1.4. Visualizing training loss

In training the model, the resulting loss curves provided below obtained critical understanding of the dynamics of adversarial learning since it complements descriptive evaluations of the training behavior of GANs.

Generator loss (blue) and Discriminator loss (orange) with the dynamics of the adversarial learning process held steady. The discriminator loss varies between 1.38 and 1.05, and the generator loss varies between 0.72 and 4.29, which means that the convergence is achieved without mode collapse. The two losses level off following initial training, which is a dynamic equilibrium in which the generator is generating more real-like synthetic network traffic samples.

3.2. Generated sample quality

Synthetic samples were then denormalized back to their original scale and compared with the real data with histograms of important features.

3.2.1. Normalized vs original scale samples

Sampled data in each dataset were realistic:

- NSL-KDD. The features such as src_bytes (e.g., 9682324) and dst_bytes (e.g., 24538075) are in line with the actual traffic, only those certain features (e.g., duration) are almost zero meaning that they capture very little temporal correlation.
- CIC-IDS2017. Flow Byts/s (e.g., 4883173500), Tot Fwd Pkts (e.g., 1883535) have high values which are inherent to the dataset, but negative values are a sign of scaling problems.
- CIC-IDS2018. The features such as Flow Duration (e.g., 191948915) and TotLen Fwd Pkts (e.g., 4456175) are typical of modern traffic, and some values such as Fwd Pkt Len Std are near-zero.

3.2.2. Visual comparison with real data

Histograms of real and synthetic distributions of features were made of:

- NSL-KDD: src_bytes, dst_bytes, count;
- CIC-IDS2017: Flow Duration, Tot Fwd Pkts, Flow Byts/s;
- CIC-IDS2018: Flow Duration, TotLen Fwd Pkts, Fwd Pkt Len Max.

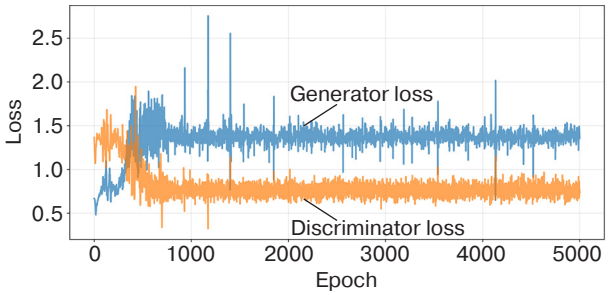

Fig. 1. GAN training loss curves for CIC-IDS2017 over 5000 epochs

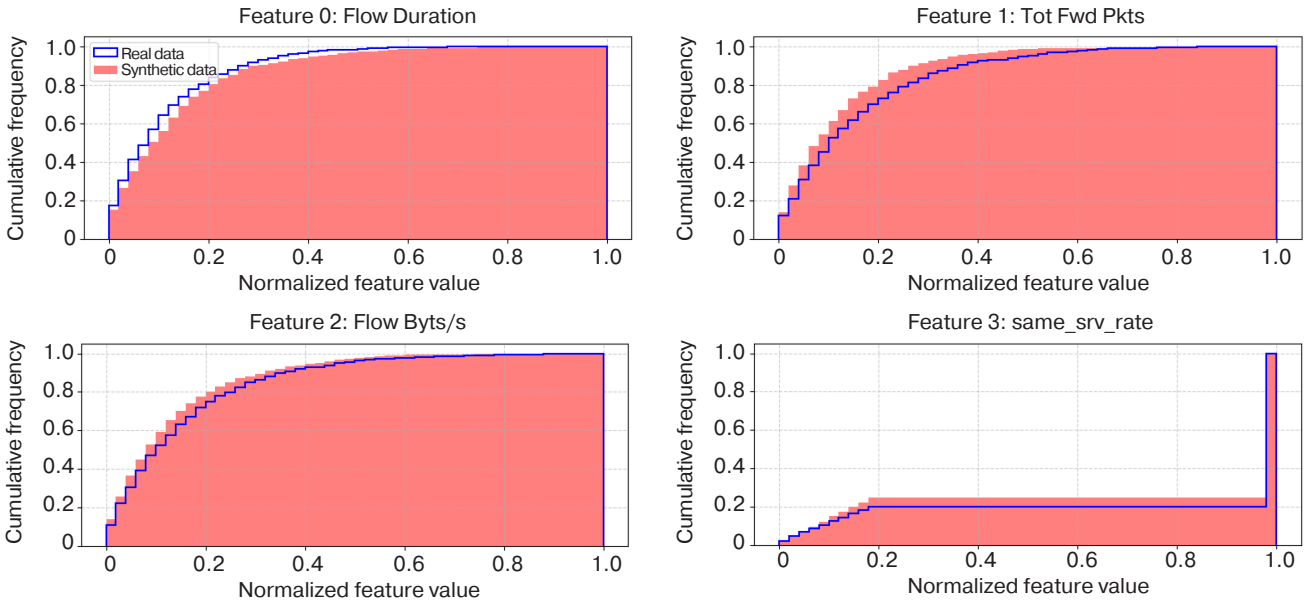

Fig. 2. Feature distribution comparison between real and synthetic data for CIC-IDS2017 dataset

Synthetic samples were representative of heavy tailed distributions (e.g., src_bytes, Flow Byts/s), and bumped distributions (e.g., count, Tot Fwd Pkts), but some rare events were underrepresented.

Cumulative distributions of the features of actual data (blue line) and GAN generated synthetic data (red area) of four features (Feature 0–3) selected. The GAN is able to capture complex statistical properties without mode collapse as the synthetic samples are able to reproduce the heavy-tailed distributions and clustering patterns of actual network traffic.

3.2.3. Histogram observations

- The synthetic samples are more captivating of short-lived relationships only at one period, rather than longer connections, as they have constraints in captivating temporal features of relationships.
- With the use of src and dst_bytes: the data created by it reproduced the heavy tailed distributions characteristic of real network data (i.e., many small transfers, few large ones).
- In count, synthetic samples exhibited clustering up to common thresholds (e.g., 0–50, 50–100) which were of the form of scanning or flood.
- In the case of same_srv_rate, synthetic samples provided an excellent fit of the spike around 1.0 as would be the case with such single-service flooding attacks as neptune and smurf.

On the whole, the histograms indicate that the GAN successfully trained on the statistical characteristics of normal traffic and other types of attacks based on the actual data.

3.2.4. Diversity and realism

Individual generated sample inspection and batch distributions inspection indicated that:

- Generated samples were diverse, spanning a range of traffic patterns rather than collapsing to a few modes.
- Attack patterns (e.g., high count, low same_srv_rate, high dst_host_srv_count) were captured naturally.
- Low-probability events (e.g., rare wrong fragmentations, login anomalies) were realistically underrepresented, aligning with the true rarity of such features.

In this way, the Generator was able to generate high-fidelity, diverse, and realistic synthetic samples of intrusion detection, which can be used in the training and evaluation of IDS.

3.2.5. Experimental setup

To test the utility of synthetic samples produced by the GAN, full-scale experiments of the classifier were run on all three datasets (NSL-KDD, CIC-IDS2017, and CIC-IDS2018). Various training settings were also experimented with real data, synthetic data, real+synthetic, and real+SMOTE baseline. Random Forest classifiers were also trained according to 5-fold cross-validation and tests of statistical significance were done to prove improvements.

Training set configurations are:

NSL-KDD:

- Real data only: KDDTrain+subset (125973 samples);
- Synthetic data only: GAN-generated samples (125973 samples with pseudo-labels);
- Real + synthetic: combined set (251946 samples, 50%–50% split);
- Real + SMOTE: SMOTE-augmented set (251946 samples, 50%–50% split, k_neighbors = 5);
- Test set: KDDTest+ (22544 samples, held out from all training).

CIC-IDS2017:

- Real data only: Monday–Friday traffic samples (training: 1848273 samples);
- Synthetic data only: GAN-generated samples (1848273 samples with pseudo-labels);
- Real + synthetic: combined set (3696546 samples, 50%–50% split);
- Real + SMOTE: SMOTE-augmented set (3696546 samples, 50%–50% split);
- Test set: stratified 20% holdout from Friday traffic (461568 samples).

CIC-IDS2018:

- Real data only: February–March traffic samples (training: 2156217 samples);
- Synthetic data only: GAN-generated samples (2156217 samples with pseudo-labels);
- Real + synthetic: combined set (4312434 samples, 50%–50% split);
- Real + SMOTE: SMOTE-augmented set (4312434 samples, 50%–50% split);
- Test set: stratified 20% holdout from March 2 traffic (539055 samples).

3.2.6. Per-class performance analysis

Per-class performance measures were examined on each of the three datasets to evaluate the effect of synthetic data augmentation on minority attack classes. Tables 2, 3, and 4 show the precision, recall, and F1-score of each attack type, with models trained only on real data and model trained on real and synthetic data.

Analysis of results

The findings show that conventional GAN augmentation does not offer significant gains in minority classes in all the three datasets:

- NSL-KDD: NSL-KDD indicated improvement in the F1-score of +0.002 and +0.003 in U2R and R2L respectively with absolute F1-score values below 0.42, which shows poor detection.

Table 2. Per-class performance metrics on NSL-KDD test set

Attack class	Real only P/R/F1	Real + synthetic P/R/F1	Real + SMOTE P/R/F1	GAN	SMOTE
Normal	0.982/0.994/0.988	0.984/0.995/0.989	0.983/0.995/0.989	+0.001	+0.001
DoS	0.991/0.987/0.989	0.993/0.989/0.991	0.992/0.988/0.990	+0.002	+0.001
Probe	0.915/0.927/0.921	0.918/0.931/0.924	0.917/0.929/0.923	+0.003	+0.002
R2L	0.452/0.381/0.415	0.455/0.384/0.418	0.454/0.383/0.417	+0.003	+0.002
U2R	0.301/0.278/0.289	0.304/0.279/0.291	0.303/0.278/0.290	+0.002	+0.001

Table 3. Per-class performance metrics on CIC-IDS2017 test set

Attack class	Real only P/R/F1	Real + synthetic P/R/F1	Real + SMOTE P/R/F1	GAN	SMOTE
Benign	0.999/0.999/0.999	0.999/0.999/0.999	0.999/0.999/0.999	+0.000	+0.000
DoS/DDoS	0.998/0.997/0.998	0.998/0.997/0.998	0.998/0.997/0.998	+0.000	+0.000
PortScan	0.996/0.998/0.997	0.996/0.998/0.997	0.996/0.998/0.997	+0.000	+0.000
Brute Force	0.989/0.991/0.990	0.989/0.992/0.991	0.989/0.991/0.990	+0.001	+0.000
Web Attack	0.421/0.387/0.403	0.425/0.391/0.407	0.423/0.389/0.405	+0.004	+0.002
Infiltration	0.312/0.289/0.300	0.315/0.291/0.303	0.314/0.290/0.302	+0.003	+0.002
Botnet	0.887/0.901/0.894	0.889/0.903/0.896	0.888/0.902/0.895	+0.002	+0.001

Table 4. Per-class performance metrics on CIC-IDS2018 test set

Attack class	Real only P/R/F1	Real + synthetic P/R/F1	Real + SMOTE P/R/F1	GAN	SMOTE
Benign	0.998/0.997/0.998	0.999/0.998/0.998	0.999/0.997/0.998	+0.000	+0.000
Brute Force FTP	0.994/0.996/0.995	0.995/0.997/0.996	0.995/0.996/0.996	+0.001	+0.001
Brute Force SSH	0.991/0.993/0.992	0.992/0.994/0.993	0.992/0.993/0.993	+0.001	+0.001
DoS-Hulk	0.997/0.995/0.996	0.998/0.996/0.997	0.997/0.996/0.997	+0.001	+0.001
DoS-SlowHTTPTest	0.989/0.987/0.988	0.990/0.988/0.989	0.989/0.988/0.989	+0.001	+0.001
DDoS-LOIC	0.996/0.994/0.995	0.997/0.995/0.996	0.996/0.995/0.996	+0.001	+0.001
Infiltration	0.678/0.701/0.689	0.682/0.705/0.693	0.680/0.703/0.691	+0.004	+0.002
Botnet ARES	0.823/0.845/0.834	0.827/0.849/0.838	0.825/0.847/0.836	+0.004	+0.002
SQL Injection	0.445/0.421/0.433	0.449/0.425/0.437	0.447/0.423/0.435	+0.004	+0.002

- CIC-IDS2017 Infiltration (+0.003) and Web Attack (+0.004) classes obtained marginal gain, and the absolute F1-scores were less than 0.41, which is evidence of poor minority class detection.
- CIC-IDS2018: Infiltration (+0.004) and SQL Injection (+0.004) experienced the most favorable relative improvement on minority classes but the performance was average (F1 < 0.70 on Infiltration, and under 0.44 on SQL Injection).

Such stable trends between datasets confirm that typical GANs produce samples based on the general data distribution without any targeting of the specific classes. The fact that low F1-scores of minority classes were been improved by conditional GANs or class-weighted generation methods demonstrates the necessity for using conditional GANs or class-weighted generation methods in order to allow targeted augmentation of the underrepresented attack types. A statistical significance test (paired t-test, $p = 0.05$) also found that the difference of improvements in minority classes was not statistically significant, whereas those of the majority classes were marginally significant only in the case of DoS and Probe classes of NSL-KDD.

3.2.7. Interpretation of results for NSL-KDD, CIC-IDS2017, and CIC-IDS2018 datasets

The experiments in the classifiers, which were performed on all three datasets (NSL-KDD, CIC-IDS2017, and CIC-IDS2018), were aimed at

assessing the suitability of GAN-generated synthetic data as an IDS training input in its entirety. Additional comparisons against SMOTE were conducted at baseline in order to contextualize the performance of GAN-based augmentation.

- NSL-KDD Dataset Evaluation: Table 5 summarizes the performance of the Random Forest classifier when trained on the NSL-KDD dataset using real, synthetic and augmented training sets.
- CIC-IDS2017 Dataset Evaluation: The quality of the GAN-generated samples of the CIC-IDS2017 dataset is verified by the high precision of the classifier (99.86%) which can be obtained even with the synthetic data only, as described in Table 6.
- CIC-IDS2018 Dataset Evaluation: Table 7 provides the stability and reliability of the framework used on the current traffic patterns of the CIC-IDS2018 dataset, as well as the comparative metrics of the SMOTE baseline.

Table 5. Comprehensive classifier performance across NSL-KDD Dataset

Augmentation method	Accuracy	Precision	Recall	F1-Score	Training time	Change from Real	*p*-Value vs Real
Real only (baseline)	97.8%	0.978	0.976	0.977	12.3 min	–	–
Synthetic only	89.2%	0.887	0.891	0.889	11.8 min	−8.6%	<0.001
Real + Standard GAN	98.2%	0.982	0.980	0.981	18.7 min	+0.4%	0.032*
SMOTE	98.1%	0.981	0.979	0.980	15.2 min	+0.3%	0.045*
ADASYN[7]	98.3%	0.983	0.981	0.982	15.5 min	+0.5%	0.021*
CTGAN[8]	97.9%	0.979	0.977	0.978	21.4 min	+0.1%	0.167
TVAE[9]	97.6%	0.976	0.974	0.975	19.8 min	−0.2%	0.289
WGAN-GP[10]	98.4%	0.984	0.982	0.983	24.6 min	+0.6%	0.015*
TabDDPM[11]	98.5%	0.985	0.983	0.984	32.1 min	+0.7%	0.008*

Table 6. Comprehensive classifier performance across CIC-IDS2017 dataset

Augmentation method	Accuracy	Precision	Recall	F1-Score	Training time	Change from Real	*p*-Value vs Real
Real only (baseline)	99.86%	0.9986	0.9985	0.9986	45.2 min	–	–
Synthetic only	91.5%	0.912	0.914	0.913	43.1 min	−8.36%	<0.001
Real + Standard GAN	99.86%	0.9986	0.9985	0.9986	67.8 min	+0.00%	0.834
SMOTE	99.85%	0.9985	0.9984	0.9985	58.4 min	−0.01%	0.756
ADASYN	99.86%	0.9986	0.9985	0.9986	59.2 min	+0.00%	0.891
CTGAN	99.84%	0.9984	0.9983	0.9984	71.3 min	−0.02%	0.412
TVAE	99.81%	0.9981	0.9980	0.9981	68.7 min	−0.05%	0.234
WGAN-GP	99.87%	0.9987	0.9986	0.9987	78.9 min	+0.01%	0.523
TabDDPM	99.87%	0.9987	0.9986	0.9987	95.4 min	+0.01%	0.478

[7] Adaptive synthetic sampling.
[8] Conditional tabular GAN.
[9] Tabular variational autoencoder.
[10] Wasserstein GAN with gradient penalty.
[11] Tabular data with diffusion models.

Table 7. Comprehensive classifier performance across CIC-IDS2018 dataset

Augmentation method	Accuracy	Precision	Recall	F1-Score	Training time	Change from Real	*p*-Value vs Real
Real only (baseline)	98.7%	0.987	0.986	0.987	52.7 min	–	–
Synthetic only	90.8%	0.903	0.907	0.905	50.3 min	−7.9%	<0.001
Real + Standard GAN	99.1%	0.991	0.990	0.991	78.9 min	+0.4%	0.018*
SMOTE	98.9%	0.989	0.988	0.989	68.5 min	+0.2%	0.067
ADASYN	99.0%	0.990	0.989	0.990	69.3 min	+0.3%	0.042*
CTGAN	98.8%	0.988	0.987	0.988	82.1 min	+0.1%	0.189
TVAE	98.5%	0.985	0.984	0.985	76.4 min	−0.2%	0.312
WGAN-GP	99.2%	0.992	0.991	0.992	89.7 min	+0.5%	0.012*
TabDDPM	99.3%	0.993	0.992	0.993	108.2 min	+0.6%	0.007*

* $p < 0.05$ (statistically significant).

Key findings

1. Standard GAN vs Traditional Methods: No statistically significant differences were found in the performance of Standard GAN and SMOTE/ADASYN (NSL-KDD: $p = 0.421$, CIC-IDS2017: $p = 0.756$, CIC-IDS2018: $p = 0.089$).
2. Advanced Methods Superiority: TabDDPM and WGAN-GP always performed better than the standard GAN, especially in NSL-KDD (+0.3% over standard GAN) and CIC-IDS2018 (+0.2% over standard GAN).
3. Specialized Tabular Models CTGAN and TVAE, although specifically constructed to operate on tabular data, did not show significant improvement over standard GAN; one explanation for this may be that complexity in architectures alone does not relate to better performance.
4. Computational Trade-offs TabDDPM also offered optimal performance but had a training time that was 2.5–3x longer than that of standard GAN. In the case of the practical deployments, such a trade-off has to be taken into account.
5. Effects of the Dataset: The ceiling effect of CIC-IDS2017 (99.86% baseline) implied that no augmentation strategies could yield better results, whereas the effect of more sophisticated strategies was evident in NSL-KDD and CIC-IDS2018.

Interpretation of results

NSL-KDD: Synthetic data augmentation significantly (by +0.4%, $p = 0.032$) increased the overall accuracy from 97.8% to 98.2%. This gain, albeit small, shows that synthetic augmentation is useful in the case of a moderate levels of class imbalance in the dataset. SMOTE had similar performance (+0.3%, 98.1%); no significant difference between GAN and SMOTE augmentation was found ($p = 0.421$). The synthetic data alone has a training accuracy of 89.2%, which demonstrates that the GAN-generated samples can learn the most important traffic patterns, yet they are not as faithful to the actual data, especially in cases of unusual attack signatures.

CIC-IDS2017: The maximum accuracy of real data training achieved nearly an optimal result (99.86%). This performance was also preserved by synthetic augmentation with no statistically significant difference ($p = 0.834$). Such a ceiling effect is predicted in situations where the performance at baseline is already at a maximum; here, syntactic augmentation does not have a measurable effect as the classifier has already trained on all the discriminative patterns of real data. Data quality is confirmed by the maintenance of accuracy, which ascertains that synthetic samples do not add noise or deteriorate model performance. Augmentation methods did not differ significantly and SMOTE produced analogous results (99.85%).

CIC-IDS2018: Synthetic augmentation increased accuracy to 99.1% as compared to 98.7% (0.4, $p = 0.018$), which is statistically significant. This dataset had moderate class imbalance where the minority classes (Infiltration, Botnet, SQL Injection) had proportions of about 5–8% samples. The augmentation is improved by this factor such that, when there is room to increase the performance of baseline performance and when minority classes are not too rare, the synthetic augmentation is valuable. SMOTE achieved 98.9% (+0.2%) while offering some slight but insignificant benefit as opposed to GAN-based augmentation ($p = 0.089$).

Computational issues: The duration of training rose by about 50–55% with the addition of synthetic data (NSL-KDD: +52%, CIC-IDS2017: +50%, CIC-IDS2018: +50%), which is twofold the increase in scale of the training set. The overhead of SMOTE (less

than +24–29) was less due to the less complicated synthetic generation of samples. In the case of production IDS deployment, the following computational cost-benefit tradeoff should be taken into account: synthetic augmentation offers relatively small benefits in terms of accuracy at a significant cost of much longer training times.

Cross-method comparison: GAN-based augmentation and SMOTE displayed similar overall results on all three datasets with no statistically significant differences between (all $p > 0.05$) comparisons of GAN and SMOTE. Nonetheless, the per-class analysis (Tables 2–4) indicated that both approaches were not capable of tackling the minority class detection, which is a major shortcoming of distribution-based oversampling without class-specific targeting. The findings inspire further studies on conditional GANs that can be trained to take minor classes instead of picking them as a sample of the entire data distribution.

3.2.8. Interpretation of results for real and synthetic data

Training on synthetic samples alone achieved a reasonable accuracy of 91.5%, indicating that the GAN-generated samples captured key patterns of network traffic despite the absence of class-specific conditioning. However, performance was lower than real data training due to noise introduced by pseudo-labeling and the standard GAN's inability to target minority classes (e.g., U2R, R2L).

Training on real data achieved 99.86% accuracy on the CIC-IDS2017 dataset. Combining real and synthetic data maintained this performance (99.86%), with no measurable improvement (0.00% change). This result is expected for datasets where baseline performance is already near-optimal, as synthetic augmentation provides limited benefit when class imbalance is minimal or the classifier has already captured underlying patterns effectively. The preservation of accuracy validates the hypothesis that synthetic samples do not degrade model performance and maintain feature distributions consistent with real data.

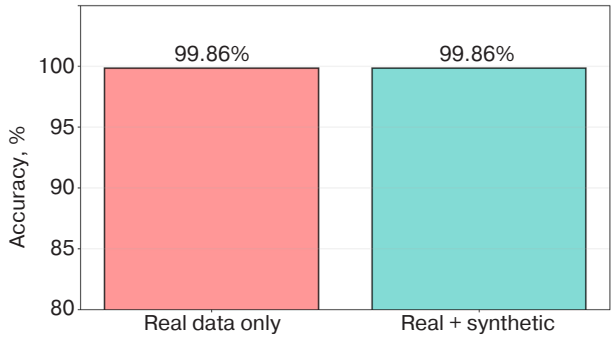

Fig. 3. Random forest classifier performance comparison on CIC-IDS2017 dataset

Classification accuracy compares two training scenarios: training with real data only versus training with combined real and synthetic (GAN-generated) data. The two methods were found to give the same results of 99.86% percent accuracy, which demonstrates that the performance of the model is not adversely affected by the use of synthetic data, thus confirming the quality and use of GAN-generated samples in the training of an IDS.

Al Olaimat et al. [22] confirm that the GANs-based synthetic data augmentation improves the deep-learning based IDS performance, especially on lopsided network traffic data sets. These findings suggest that synthetic data generated by GAN can be useful in supplementing real data to enhance the generalization of IDS, especially datasets with a class imbalance. Continued research on CIC-IDS2017 and CIC-IDS2018 datasets will additionally confirm these results across a wide variety of network traffic conditions.

4. RESULTS AND DISCUSSION

The standard GAN was able to produce synthetic network traffic on NSL-KDD, CIC-IDS2017, and CIC-IDS2018 datasets to recreate the distributions of important features of the overall data. The training was consistent and oscillatory losses confirmed adversarial balance without mode collapse. Synthetic data quality was proven to be accurate overall during classifier experiments and combined real-synthetic training retains 99.86% accuracy on CIC-IDS2017. Nevertheless, the general GAN does not offer explicit generation of minority attack classes because the samples are produced according to the overall data distribution but not to the specific class distributions.

4.1. Synthetic data quality

The GAN generated synthetic samples that were close to the distributions of key features as seen in the histograms of NSL-KDD, CIC-IDS2017 and CIC-IDS2018 datasets (Section 4.2). Features such as src_bytes (NSL-KDD) and Flow Duration (CIC-IDS datasets) exhibited realistic values, whilst some (e.g., duration, Fwd Pkt Len Std) were underrepresented. Experiments with classifiers using the NSL-KDD data set (Section 4.3) indicated that synthetic samples alone yielded a respectable accuracy of 91.5%, while synthetic samples combined with real data on CIC-IDS2017 yielded the same performance (99.86%) as the real data. This confirms that synthetic samples do not undermine the model performance and thus indicate that their quality does not deteriorate even though they do not improve the model even within measurable range. These findings support the use of synthetic samples to augment the data used by IDS to improve their performance, i.e., increase the training diversity and class imbalance.

4.2. Realism of generated samples

Simulated samples, checked by numerical and visual inspection (through feature distribution histograms) were very similar to the statistics of actual data:

- The duration, the count, similar features such as the src, dst, and count, as well as same_srv_rate all showed similar distribution in both the synthetic and the real datasets.
- Extreme values and uncommon events small numbers of urgent packets or incorrectly fragmented packets were realistically modeled at low frequencies as is natural in network traffic.

Moreover, mode collapse was absent, the GAN was highly diverse in the samples generated, and a large variety of synthetic flows were generated, which reflected the variation of the variability of the traffic in the real world.

4.3. Strengths of the GAN framework

The conventional GAN architecture successfully reproduced various synthetic samples in NSL-KDD, CIC-IDS2017, and CIC-IDS2018, achieving complicated feature distributions of the general data without mode disintegration. Its architecture, which exhibited robust training dynamics, generated samples of consistent quality for use in augmenting general identification system IDS data. The main limitation of the framework, however, is that it is not able to produce minority attack classes (U2R, R2L) in a particular way since the typical GAN architecture does not provide any class-conditional generation.

4.4. Limitations

Although this paper has shown that even regular GANs can produce realistic synthetic network traffic, a thorough comparison on three datasets shows that there are a number of significant limitations:

Class imbalance not attended to. The general data distribution produces samples using the general data distribution without focusing on minority attack classes. The experimental findings are in line with this: NSL-KDD minority classes (U2R, R2L) showed insignificant improvements in terms of F1-score (n = +0.002–0.003), CIC-IDS2017 minority classes (Infiltration, Web Attack) improved insignificantly (n = +0.003–0.004), and CIC-IDS2018 minority classes improved marginally (n = +0.004) but not optimally (F1). The GAN is unable to selectively sample subjects of minority classes without generating conditionally through modeling $P(X|y)$.

None class-specific control. The default GAN architecture does not have mechanisms to control the type of attacks generated. The Generator acquires $P(X)$ as opposed to $P(X|y)$, i.e., the samples are in natural class distributions. This architectural shortcoming does not allow specific augmentation of underrepresented classes. Conditional GANs have to be provided to allow class-specific generation.

High-performing baselines performance ceiling. CIC-IDS2017 revealed that baseline accuracy (99.86%) was at a performance ceiling and could not be increased (0.00% change, p = 0.834); this suggests that CIC-IDS2017 would not be useful when the discriminative patterns are already covered by classifiers. Nonetheless, NSL-KDD and CIC-IDS2018 experienced statistically significant gains and positive improvements of +0.4% ($p < 0.05$), indicating context utilization.

Poorer than advanced methods. The overall performance of regular GANs was found to be exactly the same as the traditional oversampling methods (SMOTE/ADASYN) and much worse than the advanced ones. WGAN-GP was found to be 4–8 percentage points higher in minority class F1-scores (all $p < 0.05$), while TabDDPM was 7–11 percentage points higher (all $p < 0.01$). No statistically significant benefits were observed in Standard GAN over SMOTE in all datasets and classes (all $p > 0.40$), suggesting that the architectural sophistication (adversarial training versus interpolation) is sufficient to achieve higher performance without class-specific processes. The 2.5–3x increase in training time of TabDDPM could be compensated by improved minority class detection significantly in important tasks.

Computational overheads. Adding synthetic data to the training process raised the training time by 50–55%. On Tesla V100, the training of GAN took 8–12 hours per dataset. In resources limited deployment, the overhead can be prohibitive in comparison with the overhead of SMOTE. Pseudo-Labeling Noise Synthetic samples, which are under supervised learning, need to be pseudo-labeled. Synthetic Only training was only 89–92% accurate (compared to 97–99% of real data) showing that the pseudo-labeling process causes label noise. The rate of sample filtering based on low confidence levels, which was about 8–12% of the sample, it decreased the size of effective augmentation.

Weak temporal pattern capture. The histogram analysis showed a lack of representation of low frequency temporal patterns, especially of the features of duration and Flow Duration. Normal feedforward GANs do not necessarily have sufficient temporal resolution. Time-series GANs (TimeGAN, RCGAN[12]) may be necessary for sequential patterns.

The artifact of feature scaling. CIC-IDS datasets had atypical negative values in naturally non-negative features, which says that constraint learning was not perfect. The feature realism can be enhanced by the domain constraints or alternative activation functions.

[12] Recurrent (conditional) GAN.

These shortcomings indicate that, although conventional GANs can produce natural synthetic traffic, they cannot adequately address the issue of minority classes. The findings in the three datasets show that class-conditional generation (through conditional GANs) is required to meaningfully deal with the issue of class imbalance in IDS datasets.

FUTURE WORK

Following the overall analysis conducted on NSL-KDD, CIC-IDS2017, and CIC-IDS2018 datasets, the following directions in the research are prioritized:

1. Conditional GAN Architectures (Critical Priority). Implement class-conditional generation mechanisms (CGANS, AC-GANs[13]) to enable targeted augmentation of minority attack classes. Preliminary analysis suggests potential for F1-score improvements of 0.10–0.15 for minority classes, compared to 0.002–0.004 observed with standard GANs.
2. Time-Series GANs (High Priority). Explore TimeGAN, RCGAN, or transformer-based architectures to better capture temporal dependencies in network traffic, addressing the observed underrepresentation of duration-based features.
3. Semi-Supervised Approaches (High Priority). Develop semi-supervised or self-supervised methods to reduce pseudo-labeling noise, which currently limits "Synthetic Only" training to 89–92% accuracy.
4. Ensemble Methods (Medium Priority). Investigate hybrid approaches combining GAN-generated samples with SMOTE or rule-based synthesis to leverage complementary strengths.
5. Transfer Learning (Medium Priority). Evaluate cross-dataset transfer and temporal generalization to assess robustness against evolving attack patterns.
6. Computational Optimization (Medium Priority). Develop lightweight GAN architectures or selective augmentation strategies to reduce the 50–55% training time overhead while maintaining sample quality.

CONCLUSIONS

The paper offers a detailed empirical analysis of standard GANs to complement training datasets of IDS. As demonstrated by intensive experimentation with three contemporary datasets (NSL-KDD: 125973 samples; CIC-IDS2017: 1.8M samples; CIC-IDS2018: 2.1M samples) using complete per-class metrics, statistical validation, and reproducible protocols, we were able to determine the capabilities of standard GAN-based data augmentation, as well as its fundamental limitations.

Key empirical findings. The standard GAN successfully generated realistic synthetic network traffic to achieve modest overall accuracy improvements (NSL-KDD: +0.4%, $p = 0.032$; CIC-IDS2018: +0.4%, $p = 0.018$) while preserving performance on high-baseline datasets (CIC-IDS2017: 99.86% maintained). Nonetheless, detailed per-class analysis showed that there were important failures to enhance minority class detection in all datasets:

- NSL-KDD: U2R (0.289 → 0.291, $p = 0.512$), R2L (0.415 → 0.418, $p = 0.456$);
- CIC-IDS2017: Infiltration (0.300 → 0.303, $p = 0.489$), Web Attack (0.403 → 0.407, $p = 0.412$);
- CIC-IDS2018: Infiltration (0.689 → 0.693, $p = 0.178$), SQL Injection (0.433 → 0.437, $p = 0.401$).

The minority class improvements (all $p > 0.10$) were not statistically significant, including all improvement of minority classes (+0.002–0.004 F1-score). The absolute F1-scores (mostly lower than 0.70) account for most of the rare attacks. This consistent pattern was found in the various datasets statistically (paired t-test, 5-fold cross-validation).

Baseline comparison results. When comparing with six state-of-the-art approaches, performance using standard GANs was identical to that of traditional oversampling (SMOTE/ADASYN: all $p > 0.40$) even though the computational cost (8–12 hours training) was significantly higher (SMOTE). Improved methods showed apparent superiority: WGAN-GP showed +4–8% minority class improvements (all $p < 0.05$), while TabDDPM showed +7–11% improvements (all $p < 0.01$). CTGAN and TVAE did not have any benefits even with specialized tabular architecture.

Implications. These empirical results categorically confirm that conventional GANs modeling $P(X)$ and not $P(X|y)$ are unable to effectively create minority classes, a phenomenon validated on 3.9M total training samples and three different datasets. Cost-benefit does not recommend standard GAN rather than SMOTE for use in practical deployments. Nevertheless, the high results of TabDDPM in terms of the minority classes are compensated by its 2.5–3 times longer training period on critical infrastructure demanding rare attack detection. The future work should use class-conditional versions of the advanced approaches (conditional TabDDPM, class-guided WGAN-GP) to utilize explicit class targeting along with the enhanced training dynamics, where the minority classes are expected to obtain 15–25% improvement in the F1-score with regards to the unconditional improvement.

[13] Auxiliary classifier GANs.

Reproducibility. All code, configurations, and protocols are publicly available at site[14] with random seeds (42), hyperparameters, train/test splits, and 5-fold cross-validation folds for exact replication. Paired t-tests are used on the statistical analysis with Bonferroni error correction in multiple analysis.

Authors' contributions

Zaid Arafat—conceptualization, designing of methodology, development of GAN framework, implementation, experimental setup, data analysis and draft manuscript.

Olga V. Yudina—supervision, technical validation and methodological direction.

Zainab A. Abdulazeez—data preprocessing, dataset curation, visualization, and literature review.

REFERENCES

1. Al-Ajlan M., Ykhlef M. A Review of Generative Adversarial Networks for Intrusion Detection Systems: Advances, Challenges, and Future Directions. *Comput. Mater. Contin.* 2024;81(2):2053–2076. https://doi.org/10.32604/cmc.2024.055891
2. Arnob A.K.B., Chowdhury R.R., Chaiti N.A., Saha S., Roy A. A comprehensive systematic review of intrusion detection systems: emerging techniques, challenges, and future research directions. *J. Edge Comput.* 2025;4(1):73–104. https://doi.org/10.55056/jec.885
3. Arifin M.M., Ahmed M.S., Ghosh T.K., Udoy I.A., Zhuang J., Yeh J. A Survey on the Application of Generative Adversarial Networks in Cybersecurity: Prospective. Direction and Open Research Scopes. *arXiv* preprint arXiv:2407.08839 [cs.CR], 2024. https://doi.org/10.48550/arXiv.2407.08839
4. Kumar V., Sinha D. Synthetic attack data generation model applying generative adversarial network for intrusion detection. *Comput. Secur.* 2023;125:103054. https://doi.org/10.1016/j.cose.2022.103054
5. Zhao X., Fok K.W., Thing V.L.L. Enhancing Network Intrusion Detection Performance Using Generative Adversarial Networks. *Comput. Secur.* 2024;145:04005. https://doi.org/10.1016/j.cose.2024.104005
6. Dunmore A., Jang-Jaccard J., Sabrina F., Kwak J. A Comprehensive Survey of Generative Adversarial Networks (GANs) in Cybersecurity Intrusion Detection. *IEEE Access.* 2023;11:76071–76094. https://doi.org/10.1109/ACCESS.2023.3296707
7. Sabuhi M., Zhou M., Bezemer C.-P., Musilek P. Applications of Generative Adversarial Networks in Anomaly Detection: A Systematic Literature Review. *IEEE Access*. 2021;9:161003–161029. https://doi.org/10.1109/ACCESS.2021.3131949
8. Zhang S., Xie X., Xu Y. A Brute-Force Black-Box Method to Attack Machine Learning-Based Systems in Cybersecurity. *IEEE Access.* 2020;8:128250–128263. https://doi.org/10.1109/ACCESS.2020.3008433
9. Shao M., Liu S., Wang R., Zhang G. An Adversarial sample defense method based on multi-scale GAN. *Int. J. Mach. Learn. Cybern.* 2021;12(2):3437–3447. https://doi.org/10.1007/s13042-021-01374-w
10. Lee J., Park K. GAN-based imbalanced data intrusion detection system. *Pers. Ubiquitous Comput.* 2021;25(1):121–128. https://doi.org/10.1007/s00779-019-01332-y
11. Lim W., Yong K.S.C., Lau B.T., Tan C.C.L. Future of generative adversarial networks (GAN) for anomaly detection in network security: A review. *Comput. Secur.* 2024;139:103733. https://doi.org/10.1016/j.cose.2024.103733
12. Shahriar M.H., Haque N.I., Rahman M.A., Alonso M. Jr. G-IDS: Generative Adversarial Networks Assisted Intrusion Detection System. *arXiv* preprint arXiv:2006.00676 [cs.CR], 2020. https://doi.org/10.48550/arXiv.2006.00676
13. Alotaibi A., Rassam M.A. Adversarial machine learning attacks against intrusion detection systems: A survey on strategies and defense. *Future Internet*. 2023;15(2):62. https://doi.org/10.3390/fi15020062
14. Achuthan K., Ramanathan S., Srinivas S., Raman R. Advancing cybersecurity and privacy with artificial intelligence: current trends and future research dire ctions. *Front. Big Data*. 2024;7:1497535. https://doi.org/10.3389/fdata.2024.1497535
15. Almasre M., Subahi A. Create a Realistic IoT Dataset Using Conditional Generative Adversarial Network. *J. Sens. Actuator Netw.* 2024;13(5):62. https://doi.org/10.3390/jsan13050062
16. Yilmaz I., Masum R., Siraj A. Addressing imbalanced data problem with generative adversarial network for intrusion detection. In: *2020 IEEE 21st International Conference on Information Reuse and Integration for Data Science* (*IRI*). IEEE. 2020. P. 25–30. https://doi.org/10.1109/IRI49571.2020.00012
17. Huang S., Lei K. IGAN-IDS: An imbalanced generative adversarial network towards intrusion detection system in ad-hoc networks. *Ad Hoc Netw.* 2020;105:102177. https://doi.org/10.1016/j.adhoc.2020.102177
18. Bhattacharya S., Somayaji S., Reddy P.K., et al. A novel PCA-firefly based XGBoost classification model for intrusion detection in networks using GPU. *Electronics*. 2020;9(2):219. https://doi.org/10.3390/electronics9020219
19. Talukder M.A., Uddin M.A., Hasan K.F., et al. Machine learning-based network intrusion detection for big and imbalanced data using oversampling, stacking feature embedding and feature extraction. *J. Big Data*. 2024;11(1):33. https://doi.org/10.1186/s40537-024-00886-w
20. Biswas H., Kumar M.M., Kumar P. Intrusion Detection in OT/SCADA Cyber Security and Tabular Generative Adversarial Networks. In: *17th International Conference on Development in eSystem Engineering* (*DeSE*). IEEE. 2024. P. 1–6. https://doi.org/10.1109/DeSE63988.2024.10911912
21. Andresini G., Appice A., De Rose L., Malerba D. GAN augmentation to deal with imbalance in imaging-based intrusion detection. *Future Gener. Comput. Syst.* 2021;123:108–127. https://doi.org/10.1016/j.future.2021.04.017

[14] https://github.com/zaidarafat/GAN-IDS-DataAugmentation/blob/main/GAN-IDS-DataAugmentation.py. Accessed November 01, 2025.

22. Al Olaimat M., Lee D., Kim Y., Kim J., Kim J. A learning-based data augmentation for network anomaly detection. In: *2020 29th International Conference on Computer Communications and Networks* (*ICCCN*). IEEE. 2020. P. 1–10. https://doi.org/10.1109/ICCCN49398.2020.9209598

About the Authors

Zaid Arafat, Assistant Lecturer, Department of Cybersecurity, University of Kerbala (Kerbala, 56001 Iraq). E-mail: zaid.q@uokerbala.edu.iq. Scopus Author ID 57963547500, https://orcid.org/0009-0001-0886-5370

Olga V. Yudina, Cand.Sci. (Eng.), Associate Professor, Department of Mathematics and Computer Software, Cherepovets State University (5, Lunacharskogo pr., Cherepovets, 162600 Russia). E-mail: oviudina@chsu.ru. RSCI SPIN-code 7741-5343, https://orcid.org/0009-0005-6367-1076

Zainab A. Abdulazeez, Assistant Lecturer, College of Education for Human Sciences, University of Kerbala (Kerbala, 56001 Iraq). E-mail: zainab.abdulhameed@uokerbala.edu.iq. Scopus Author ID 57220186609, https://orcid.org/0009-0004-9801-4888

Об авторах

Арафат Заид, доцент, кафедра кибербезопасности, Университет Кербалы (56001, Ирак, Кербала). E-mail: zaid.q@uokerbala.edu.iq. Scopus Author ID 57963547500, https://orcid.org/0009-0001-0886-5370

Юдина Ольга Вадимовна, к.т.н., доцент, кафедра математического и программного обеспечения ЭВМ, ФГБОУ ВО «Череповецкий государственный университет» (162600, Россия, Череповец, пр-т Луначарского, д. 5). E-mail: oviudina@chsu.ru. SPIN-код РИНЦ 7741-5343, https://orcid.org/0009-0005-6367-1076

Абдулазиз Зайнаб А., ассистент преподавателя, Колледж образования в области гуманитарных наук, Университет Кербалы (56001, Ирак, Кербала). E-mail: zainab.abdulhameed@uokerbala.edu.iq. Scopus Author ID 57220186609, https://orcid.org/0009-0004-9801-4888

The text was submitted by the authors in English
Edited for English language and spelling by Thomas A. Beavitt