Al-Rafidain Journal of Computer Sciences and Mathematics (RJCSM)

www.csmj.uomosul.edu.iq

Dual-GAN Framework for Adversarial Robust Intrusion Detection Systems

Zainab A. Abdulazeez

College of Education for Human Sciences, University of Kerbala, Karbala, Iraq
Email: zainab.abdulhameed@uokerbala.edu.iq

Article information

Article history:
Received 10 February, 2026
Revised 14 April, 2026
Accepted 24 April, 2026
Published 25 June, 2026

Keywords:
Intrusion Detection Systems,
Generative Adversarial Networks,
Adversarial Machine Learning,
Data Augmentation,
Network Security.

Correspondence:
Zainab A. Abdulazeez
Email:
zainab.abdulhameed@uokerbala.edu.iq

Abstract

The importance of intrusion detection systems (IDSs) is part of existing cybersecurity facilities. However, existing machine learning-based IDSs are vulnerable to adversarial attacks and evasion techniques. Generative adversarial networks (GANs) have demonstrated the possibility of using synthetic data augmentation to address the problem of imbalance in network security datasets. Nevertheless, current methods examine performance using clean datasets without considering resilience to adversarial corruption. In this study, a framework of dual GAN (DGF) is proposed, in which one of the GANs produces data to augment the data with synthetic samples, and the other GAN generates adversarial instances to improve the robustness of the model in the training process. The synthesis of synthetic data and the adversarial robustness of intrusion detection have been under-researched. The proposed framework focuses on this aspect. The experimental results on the NSL-KDD benchmark dataset indicate that DGF decreases the false-negative rates by an absolute of 1.11% (11.11% on clean data and 10.00% on evasion conditions), and its relative performance is decreased by 41.5 %–46.0% on augmentation-only baselines. The accuracy of the framework in detecting clean data reached 92.44%, and under adversarial conditions, 92.13% (only 0.31% decreased). This is better than the existing procedures, which have a reduction in accuracy of over 30% in the case of the same perturbations. These findings reveal how dual-purpose GAN architectures can be useful in designing IDSs that work in adversarial network settings.

DOI: 10.33899/rjcsm.v20i1.60660,

1. Introduction

The prevalence of complex cyber threats has elevated the significance of network intrusion detection in organizational security systems. Machine learning-based intrusion detection systems (IDSs) have become popular solutions. They can identify new attack patterns without predefined signatures. [1]. However, two interdependent issues restrict their operational efficiency. First, the training data have a severe class imbalance, in which a large percentage of benign traffic exceeds the number of malicious cases. Second, these systems are vulnerable to adversarial attacks to avoid detection. [2].

GANs have proven useful in resolving class imbalances by creating synthetic data. Architectures such as Conditional GAN (CGAN) and Wasserstein GAN with Gradient Penalty (WGAN-GP) have generated realistic synthetic attack examples. This enhances the performance of minority classes and classifiers. [3]. Recent implementations of CE-WGAN have shown improvements in identifying unusual types of attacks using improved training samples [4]. However, these approaches have major limitations. They used clean data and did not use adversarial hardening when performing their analysis.

Adversarial machine learning research has demonstrated that deep learning models for intrusion detection are susceptible to well-constructed perturbations. They result in misclassification, although they do not lose their semantics. [5]. Adversarial examples are distorted malicious traffic that is not noticed but still functions as an attack. The literature on adversarial attacks on IDS has mostly concentrated on the aspects of proving

vulnerabilities and not the creation of defenses [6], as indicated by surveys that inadequate defenses against evasion attacks with a loss of accuracy of more than 30% [7].

This study addresses this gap, which was identified in the literature on augmentation-oriented methods [3], [4], [8], [9], [10] and demonstrations of vulnerabilities [5], [6], [11], [12], [13], [14] with a dual-generative adversarial network (DGF) framework that incorporates both data augmentation and adversarial robustness. The framework comprises two synchronized GAN modules. An augmentation GAN (A-GAN) is used to generate synthetic attack samples to equalize the classes. The adversarial GAN (Adv-GAN) generates adversarial samples to reinforce the detection model, which is adversarially trained. The two-fold architecture is a step towards using synthetic data generation and adversarial defense in network security, offering a single framework to solve these problems separately.

The main contributions of this study are as follows.

1- An original dual-GAN model that combines data augmentation and adversarial robustness training is proposed in this study.

2- This method involves a thorough evaluation to determine the IDS performance in clean and adversarial environments.

3- Evidence of a decrease in the false-negative rate with evasion of 1.11 percentage points absolute (from 11.11% to 10.00%) under evasion compared to clean data, with relative reductions and relative reductions of 41.5–46.0% under augmentation-only baselines that are above the 15 %–25% target range.

4- Insights into the synergistic relationship between synthetic data quality and adversarial resilience in intrusion detection systems

The remainder of this paper is organized as follows. In Section 2, a comprehensive review of GAN-based IDS augmentation and adversarial machine learning in network security is presented. Section 3 describes the proposed methodology, architectural details, and training processes. Section 4 describes the experimental design, data, evaluation measures, and baselines. Section 5 presents the experimental results and a detailed analysis. The limitations of this study are discussed in Section 6. Section 7 concludes the paper and provides recommendations for future research.

2. Related Work

2.1. GAN-Based Data Augmentation for Intrusion Detection

Generative adversarial networks (GANs) have been widely researched for intrusion detection to overcome class imbalance. Shahriar et al. [15] did a systematic survey of GAN in cybersecurity. One of the applications they identified was data augmentation. Their comparison showed that the performance of the classifier on minority attack classes was regularly improved using synthetic samples produced by GANs. The F1-scores for rare attacks increased by 5%–15%.

Ring et al. studied flow-based network intrusion detection datasets [11]. They emphasized the constant problem of class imbalance, in which the attack samples were less than 1% of the observations. Such a skew would favor the majority of benign classifiers. To produce a sample of attacks that is representative and diverse, the authors suggested augmentation techniques.

The IDSGAN framework was proposed by Lin et al. [8]. It applies GANs to generate adversarial malicious traffic that is not detected. Although this work focused on illustrating vulnerabilities, it demonstrated the capability of GANs to create semantically unharmed attack traffic. It has been used in future studies on the offensive and defensive applications of GANs in network security.

The gradient penalty conditional WGAN has been successfully applied to a set of augmentation techniques. Klinkhamhom et al. [9] showed that CWGAN generates more fidelity samples compared to other GAN variants to original attack distributions. It improved the underrepresented U2R and R2L attacks by 12% on the NSL-KDD dataset.

Liu et al. [10] used privacy-preserving intrusion detection by combining GAN-generated information with federated learning. Their strategy addressed the lack of data in distributed deployments without loss of privacy. However, it did not assess adversarial robustness.

Xu et al. [16] have suggested a GAN intrusion detection scheme that employs the discriminator as the sensor. This is an innovation that allows simultaneous data augmentation and classification. It has state-of-the-art results in CIC-IDS-2017; however, it was not tested under adversarial conditions.

Such works are useful for addressing the issue of class imbalance; however, they ignore the adversarial robustness aspect of defense, leaving a gap that our framework fills by combining augmentation with defense mechanisms. A systematic review of the literature by Arafat et al. [17] provides an in-depth analysis of different types of GAN algorithms, including DCGANs, WGANs, and CGANs, as applied to cybersecurity solutions, revealing the effectiveness of this type of architecture in enhancing the accuracy of IDS detection by a factor of up to 25.

2.2.Recent Advancements in Transformer-Based IDS

Recently, transformer architectures have been used in IDS to model long-range network traffic dependencies. These overcome the constraints of conventional GAN-based augmentation. For example, Rakkini et al. [18] introduced a transformer-based deep federated learning method for a privacy-preserving IDS. It is highly accurate in a distributed setting and addresses the issue of class imbalance through the generation of synthetic data. Similarly, Nalayini et al. [19] introduced an adaptive transformer-based quantum IDS for software-defined networks (SDNs). It combines quantum-informed feature selection and federated learning to increase resistance to changing threats. A survey by Jing et al. [12] highlighted the use of transformers and large language models (e.g., BERT, GPT) in efficient IDS. It records hybrid CNN/LSTM-transformer models that have better anomaly detection capabilities. These developments indicate possible synergies with our two-GAN-based hybrid models; however, they do not generally use adversarial training, highlighting the opportunity that DGF is filling. DGF performs better or similarly (on CIC-IDS-2017) to recent transformer-based IDS on clean data [18], [19].

2.3.Adversarial Machine Learning in Network Security

Extensive studies on adversarial examples have been conducted in computer vision. Gradient-based perturbations are a classical method [13]. Traffic has distinct and limited characteristics that introduce unique challenges in network intrusion detection for attack transferability.

Alotaibi and Rassam [7] surveyed adversarial attacks on network intrusion detection systems (IDSs based on machine learning. Their analysis involved white, black, and gray-box threat models. Evasion attacks are the most viable threats to deployed systems. Existing defenses are insufficient, and most of them have accuracy levels below 30% owing to targeted perturbations.

Qiu et al. [14] came up with a network traffic classification-based adversarial attack architecture. This proves that small distortions in the flow characteristics can lead to misclassification and leave the traffic valid. This highlights the importance of adversarial examples in network security, which still have functional equivalence to the original attacks.

Xiong et al. [20] investigated the use of adversarial training as a defense on intrusion detection. This enhances the gradient-based attack resistance. However, the use of predefined perturbations restricts the flexibility of new evasion. This study confirms the validity of adversarial training but requires complex systems.

Debicha et al. [21] subjected the deep learning-based intrusion detector resilience to adversarial conditions. Partial protection was provided by ensemble methods and input preprocessing. No strategy was entirely protective, encouraging the development of more robust approaches.

To solve the problem of malware detection, Wang et al. [22] suggested a GAN-based adversarial training algorithm. The examples generated by the GAN significantly increased the resistance of the classifier compared to conventional augmentation. Although it is centered on malware, it proposes that GAN adversarial training can be useful for security activities such as intrusion detection.

This literature mainly reveals the weak points [5], [6], [11], [12], [13], [14] without systematic countermeasures that also respond to imbalance; therefore, our combined dual-GAN solution is theoretically necessary.

2.4.Recent Advancements in Adaptive Adversarial Defenses

Post-2023 research emphasizes adaptive defenses that evolve with threats beyond static GAN training. Nguyen et al. [23] have conducted a review of adversarial attacks and defense in AI-powered IDS. They suggested adaptive mechanisms based on reinforcement learning to overcome dynamic evasion. In an ML-based IDS, Ennaji et al. [24] proposed a behavior-based defense against black-box attacks. It reuses adversarial training to learn incremental detection with 95–98% robust accuracy. Although promising, these defenses usually assume balanced data and fail to incorporate augmentation, indicating the gap that our framework fills.

3. Methodology

3.1.Framework Overview

The proposed Dual-GAN Framework (DGF) contains three essential elements in the training pipeline. These include an Augmentation GAN (A-GAN) of synthetic attack samples, an Adversarial GAN (Adv-GAN) of adversarial examples, and a classifier network that acts as an intrusion detection model. The architecture solves the problem of class imbalance using A-GAN augmentation. It also addresses adversarial vulnerability through Adv-GAN robust training.

The framework operates in two coordinated phases. A-GAN is used during augmentation, where underrepresented attack categories are created using artificial samples. This increases the training set and enhances class balance. During the robustness stage, the Adv-GAN creates adversarial instances of the original and augmented samples. These were added to the training of the classifiers through an adversarial procedure. This two-sided strategy guarantees that the resulting IDS has enhanced minority representation in classes and resistance to evasion.

A-GAN and Adv-GAN use the Wasserstein GAN with gradient penalty (WGAN-GP). WGAN-GP was chosen as an alternative to vanilla GAN or any of its variations because it provides theoretical benefits in terms of training stability and mode collapse prevention. WGAN-GP optimizes using the Wasserstein distance metric, unlike vanilla GANs, which have unstable gradients and vanishing issues [3], that gives more stable and reliable optimization. The gradient penalty imposes Lipschitz continuity, which further stabilizes the training of high-dimensional network traffic data that are difficult to stabilize [8]. Past studies on the augmentation of IDS using empirical evidence [4], [8] have shown that WGAN-GP has better sample fidelity than other methods, such as DCGAN or vanilla CGAN; thus, it is best at creating realistic samples of attacks with the least artifacts that may compromise robustness.

Assume that $D = \{(x_i, y_i)\}_{i=1}^{N}$ is the original training data where x_i is a network flow feature vector and $y_i \in \{0, 1, \ldots, K\}$ is the attack type (0 benign traffic, 1-K different attack types). The trained framework provides a superior and adversarially resilient classifier $C: X \rightarrow Y$ that retains high detection rates for clean and adversarially perturbed inputs.

3.2.Loss Functions

The total loss of Adv-GAN is the sum of adversarial generation, minimization of perturbation, and features: $L_{Adv} = \lambda . L_{adv} + \mu . L_{gp} + \nu . L_{feat}$, where L_{Adv} is the adversarial misclassification loss, L_{gp} is the gradient penalty, and L_{feat} enforces domain-specific constraints on generated perturbations. The coefficients were tuned via a grid search on a 20% validation split from NSL-KDD, with search ranges λ= [0.5,2], μ= [5,20], ν= [0.05,0.5]. The optimal values were λ=1 (balancing adversarial strength), μ=10 (standard for WGAN-GP stability), and ν=0.1 (ensuring valid traffic features without over-constraining generation).

3.3.Augmentation GAN Architecture

The A-GAN component has a Wasserstein GAN based on a Gradient Penalty (WGAN-GP) architecture conditioned on the attack class labels. Conditional formulation allows specific types of attacks to be generated, providing fine-tuning of the augmentation of severely underrepresented types. The noise sample is a concatenation of a one-hot encoded class label c and a noise vector z, sampled by a standard normal distribution, and input to the generator G_A to obtain synthetic samples $x_{syn} = G_A(z, c)$.

The generator architecture is composed of fully connected 128–256–512–n feature layers with the dimensionality of the network flow feature vector. Intermediate layers are batched normalized and used with Leaky ReLU with a negative slope of 0.2, and the final layer is applied with the tangent hyperbolic activation to generate features within the normalized feature range of [-1, 1].

The critic D_A has a symmetric structure of n features-512-256-128-1 layers with no batch normalization as it is recommended in the WGAN-GP implementations. Layer normalization was used as an alternative to stabilize the training. The Wasserstein gradient penalty loss is calculated as follows:

$$L_D = \mathbb{E}[D_A(x_{syn}, c)] - \mathbb{E}[D_A(x_{real}, c)] + \lambda_{gp}\mathbb{E}[(||\nabla_{\hat{x}} D_A(\hat{x}, c)||_2 - 1)^2]$$

where x̂ is considered as interpolated samples between real and generated samples, and calculated as $\hat{x} = \varepsilon x_{real} + (1 - \varepsilon) x_{syn}$ with $\varepsilon \sim U(0,1)$, and $\lambda_{gp} = 10$ is the gradient penalty coefficient. The generator loss is:

$$L_G = -\mathbb{E}[D_A(G_A(z, c), c)]$$

The conditional formulation uses an auxiliary classifier that forecasts the class of attack of the real and generated samples and ensures semantic consistency of the generated attacks. For both critic- and generator-goals, an auxiliary classification loss is introduced:

$$L_{aux} = \mathbb{E}[-\log P\,(c|x_{real})] + \mathbb{E}\big[-\log P\,\big(c\big|x_{syn}\big)\big]$$

The auxiliary classification loss is added to the main Wasserstein losses (Equations (1) and (2) for the critic and generator, respectively, as $L_{total} = L_{wgan} + L_{aux}$.

3.4.Adversarial GAN Architecture

The Adv-GAN component is specifically designed to generate adversarial perturbations that cause the classifier to misclassify attack samples as benign, while maintaining traffic validity constraints. Unlike the A-GAN, which generates complete samples from noise, the Adv-GAN produces perturbation vectors δ that are added to the original attack samples x_{attack} to create adversarial examples $x_{adv} = x_{attack} + \delta$.

The perturbation generator G_P is used by inputting the original attack sample x_{attack} and generating a bounded perturbation:

$$\delta = \varepsilon \cdot tanh(G_P(x_{attack}))$$

where ε is the maximum magnitude of the perturbation, and tanh is used to limit the perturbations to the range [-ε, ε]. This ensures that adversarial examples are maintained within the data feature ranges and that they maintain the attack semantics. The generator architecture uses an encoder–decoder architecture along with skip connections to preserve feature correlations.

The encoder comprises fully connected layers, where n is the number of features, 256-128-64 are the dimensions of the layers, and finally, there is a batch normalization and a Leaky ReLU. The decoder has the same structure, and its

size is $64 - 128 - 256 - n_{\text{features}}$ with skip connections between similar encoder layers to retain the fine-grained feature details. The last layer uses tanh activation multiplied by ε.

The Adv-GAN training objective incorporated three loss components.

1- Adversarial Loss (L_{adv}): Encourages classifier C to predict the benign class for adversarial examples: $L_{\text{adv}} = \mathbb{E}[-\log C(x_{adv})_{benign}]$

2- Perturbation Loss (L_{pert}): Penalizes excessive perturbation magnitudes to maintain validity: $L_{pert} = \mathbb{E}[||\delta||_2]$

3- GAN Loss (L_{GAN}): This ensures that the adversarial examples are indistinguishable from legitimate traffic: $L_{GAN} = \mathbb{E}[\log D_P(x_{attack})] + \mathbb{E}[\log(1 - D_P(x_{adv}))]$

The combined generator objective is as follows:

$$L_{total} = \alpha \cdot L_{adv} + \beta \cdot L_{pert} + \gamma \cdot L_{GAN}$$

with $\alpha = 1.0, \beta = 0.5,$ and $\gamma = 0.3$ being the weighting coefficients optimized on a validation datapoint.

The discriminator D_P is trained to differentiate between original attack samples and adversarial examples by the use of standard binary cross-entropy loss:

$$L_{D_P} = -\mathbb{E}[\log D_P(x_{attack})] - \mathbb{E}[\log(1 - D_P(x_{adv}))]$$

3.5.Classifier Network

C is the network C of classifiers used as the intrusion detector model to be trained to differentiate between benign traffic and various types of attacks. The deep neural network structure is used with fully connected layers of size $n_{features} - 512 - 256 - 128 - 64 - (K + 1)$ where $K + 1$ is the number of classes including the benign class. Batch normalization and ReLU activation were used after each hidden layer. A dropout probability of 0.3 was used after every hidden layer to avoid overfitting. Multiclass classification was performed using softmax activation in the output layer.

The classifier is trained using standard cross-entropy loss on the augmented training set, as well as adversarial training loss on the Adv-GAN-generated samples. Where; $D_{aug} = D \cup D_{syn}$ is the augmented training set of the original samples and A-GAN-generated samples. The loss function of the classifier is

$$L_C = \mathbb{E}_{(x,y)\in D_{aug}}[-\log C(x)_y] + \lambda_{adv} \cdot \mathbb{E}_{x_{attack}\in D_{attack}}[-\log C(x_{adv})_{Y_{attack}}]$$

Where; D_{attack} is the attack example of the augmented data, and $x_{adv} = x_{attack} + G_P(x_{attack})$ is the adversarial example, y_{attack} is the actual attack label, and λ_{adv} is the strength of the adversarial training. The second term prompts the classifier to correctly recognize adversarial examples as attacks despite the perturbations of the Adv-GAN.

3.6.Training Procedure

The model used a three-phase training process. The A-GAN is trained in Stage 1, but not retrained in Stage 3; the generated data in Stage 1 is fixed and used for Stage 2 and Stage 3. This guarantees that the augmented dataset remains the same as the classifier is trained.

Stage 1: A-GAN Pre-training. The original dataset D is fed to the A-GAN until convergence, which usually takes 200 epochs. The critic receives updated 5 times in accordance with the WGAN-GP suggestions with each update of the generator. Minority attack classes are synthetically created to obtain a desired ratio between classes, so that their augmented dataset is D_{aug}.

Stage 2: Classifier Pre-training. C is a classifier trained with the common cross-entropy loss and 100 epochs on D_{aug} with to define the baseline detection capability. It is used as a pre-trained classifier as the target model during Adv-GAN training.

Stage 3: Joint adversarial training. The Adv-GAN and classifier were jointly trained in an alternating manner. For each training iteration,

- The Adv-GAN generator G_P was updated to generate perturbations that maximized misclassification and minimized detectability.
- The Adv-GAN discriminator D_P is trained to differentiate between real and adversarial samples.
- Classifier C is trained on a combined batch of clean samples of the D_{aug} set and adversarial examples of G_P.

The joint training proceeds until 150 epochs, and early stopping is performed depending on the validation performance in adversarial settings. The coefficient λ_{adv} of adversarial training was annealed with values between 0.1 and 1.0 during the initial 50 epochs to enable progressive adjustment to more effective adversarial examples.

3.7.Feature Constraint Mechanism

The characteristics of network traffic have domain-related limitations that should be maintained in both synthetic and adversarial samples. Validity is provided by the following feature constraint mechanism:

1- Categorical Features: While features of discrete categories (e.g., protocol type and service) are not permitted to have invalid values, such as one-hot encoding and argmax selection for instance, an adversary does not convert a "protocol type" from

TCP to an invalid number (say 999) by using hard constraints such as one-hot encoding and argmax selection.

2- Non-negative Features: Packet and byte counts are limited to the use of non-negative values using the ReLU activation or absolute values.

3- Bounded features: Bounded features with known bounds (e.g., port numbers between [0, 65535]) are generated with bounds clipped to legal ranges.

4- Correlation Preservation: Feature correlations observed in the training data are promoted by using an auxiliary correlation loss: $L_{corr} = ||\sum_{syn} - \sum_{real}||_F$ where Σ is the feature correlation matrix and $||\cdot||_F$ denotes the Frobenius norm.

For adversarial perturbations, more constraints are required to make sure that perturbations do not result in invalid or obvious anomalies. We do not change the features that the adversary cannot modify (e.g., timestamp-related features) and the magnitude of the perturbation is related to the variance of the features to avoid the creation of outliers. Hard constraints (such as clipping or masking) are preferred to soft constraints (such as correlation loss) to ensure validity.

4. Experimental Setup

4.1.Datasets

An experimental evaluation was conducted using the Network Security Lab KDD (NSL-KDD) benchmark dataset for network intrusion detection.

NSL-KDD: This is a modification of the KDD Cup 1999 dataset, which resolves the problems of redundancy and class imbalance that were present in the original. The dataset has 125,973 training cases and 22,544 test cases in five classes, namely, Normal, DoS (Denial of Service), Probe, R2L (Remote to Local), and U2R (User to root) cases. The sample data have an extreme case of class imbalance, in which U2R is represented by a mere 0.04% of the training samples.

CIC-IDS-2017: A modern dataset produced by the Canadian Institute of Cybersecurity, consisting of realistic network traffic recorded over five days. The sample consists of benign traffic and 14 types of attacks, which are divided into the following main categories: DoS/DDoS, brute force, web attack, infiltration, botnet, and port scan. The dataset contains approximately 2.8 million samples after preprocessing was performed to eliminate duplicate and incomplete data, and the classes are significantly imbalanced in the attack categories.

After preprocessing, the class distribution is as follows: BENIGN (2,273,097 samples), DoS Hulk (231,073), PortScan (158,930), DDoS (128,027), DoS GoldenEye (10,293), FTP-Patator (7,938), SSH-Patator (5,897), DoS slow-loris (5,796), DoS slow-http-test (5,499), and other minority classes with fewer samples.

4.2.Data Preprocessing

Both datasets were processed in a standardized manner.

1- Feature selection: Features that were constant or almost constant (variance < 0.01) were eliminated. In the case of NSL-KDD, there were 41 features. In CIC-IDS-2017, 78 features remained after eliminating identifiers and highly correlated features (correlation > 0.95).

2- Normalization: Min-max scaling was used to normalize the numerical features in the range of [-1, 1] to facilitate the GAN training with tanh output activations.

3- Encoding: Categorical values were one-hot encoded, and the number of dimensions encoded was added to the number of features.

4- Train-Test Split: In CIC-IDS-2017, the data are split into 80% and 20% to train and test the model, respectively, maintaining the time order to avoid information leakage by following a temporal split according to the flow timestamps (first 80% of chronologically ordered flows for training and the last 20% for testing). The NSL-KDD dataset uses a pre-specified train-test split.

4.3.Evaluation Metrics

The following metrics were used for the performance evaluation:

1- Accuracy: General percentage of accurate samples.

2- Precision, recall, and F1-score: These were calculated on a per-class basis and averaged across all classes to correct the influence of class imbalance.

3- False-negative rate (FNR): Portion of the samples of attacks falsely labeled as benign, which is vital in security applications.

4- Detection rate (DR): The rate at which samples of attacks are correctly recognized, which is the same as the attack class recall.

5- Area Under the ROC Curve (AUC): A cumulative parameter of classification in the form of decision thresholds.

In the case of adversarial evaluation, the following measures were added:

6- Adversarial accuracy, Adversarial classification accuracy.

7- Robustness Degradation: Percentage change in accuracy in adverse conditions over clean conditions.

8- Attack success rate (ASR): The ratio of adversarial instances that are successfully avoided.

4.4.Baseline Methods

The suggested DGF was contrasted with the no baseline strategy.

1- Standard DNN: Deep neural network classifier trained on original imbalanced data without data augmentation or adversarial training.

2- SMOTE-DNN: Classifier of the DNN on data augmented with the synthetic minority over-sampling technique (SMOTE) [25].

3- GAN-Aug: A DNN classifier trained on data augmented by a standard conditional WGAN-GP, which is an augmentation-only method.

4- Adv-Train: A DNN classifier adversarially trained with adversarial examples generated by FGSM [13], which is an adversarial defense-only model.

5- IDSGAN-Defense: Defense adaptation of the IDSGAN framework [20] in which robust training is performed using GAN-generated adversarial examples.

4.5.Adversarial Attack Scenarios

The adversarial attack on the model's robustness was assessed as follows:

1- Fast Gradient Sign Method (FGSM): This method is a gradient-based perturbation that uses, $\varepsilon \in \{0.01, 0.05, 0.1\}$ which indicates the magnitude of the perturbation. The test was run with ε=0.1 (weak tie of subtle evasions) and extrapolated to ε=0.3 (extreme cases, simulation of even stronger attacks).

2- Projected gradient descent (PGD): 20 iterations of the gradient attack, step size α = ε/10, and randomized initialization. Evaluated for ε=0.1 to 0.3.

3- Carlini and Wagner (C&W): Attack on the L2 norm using an optimization-based attack and confidence parameter κ = 0.

4- Feature-constrained Attack: An attack that honors network traffic feature constraints and alters features that the attacker can adjust.

5- Adv-GAN attack: This is a black-box attack that uses an independently trained Adv-GAN to measure transferability.

Extreme cases were also checked using higher ε values (up to 0.3) to ensure that the robustness of the DGF did not deteriorate as the accuracy decreased to less than 5% at ε=0.3, compared with the baselines of more than 40%.

4.6.Implementation Details

The following hyperparameters were used to implement the framework using PyTorch 2.0.

- Adam with learning rate 0.0002, β1 = 0.5, β2 = 0.999
- Batch size: 256
- A-GAN training epochs: 200
- Classifier pre-training steps: 100.
- Epochs of joint adversarial training: 150 epochs.
- Gradient penalty coefficient λ_{gp}: 10
- Adversarial perturbation bound ε: 0.1
- Adversarial training coefficient λ_{adv}: annealed from 0.1 to 1.0
- Augmentation target ratio: minority classes augmented to 10% of majority class size

The experiments were performed on a workstation with an NVIDIA RTX 3090 (24GB) internal memory, AMD Ryzen 9 5950X CPU, and 64 GB internal RAM. The experiments were repeated five times with various random seeds, and the reported results are expressed as the mean ± standard deviation.

5. Results

5.1.Performance on Clean Data

Table 1 presents the classification measures for the clean NSL-KDD data. The DGF has an accuracy of 92.44%, which exceeds the baseline. Architectural advantage It decreases the false-negative rate (FNR) by 1.11% relative to GAN-Aug.

Table 1. Classification Performance on NSL-KDD (Clean Data)

Method	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	FNR (%)
Standard DNN	81.2 ± 0.8	76.4 ± 1.2	71.8 ± 1.5	73.9 ± 1.3	28.2 ± 1.5
SMOTE-DNN	84.6 ± 0.6	80.2 ± 0.9	78.5 ± 1.1	79.3 ± 0.9	21.5 ± 1.1
GAN-Aug	86.3 ± 0.5	82.7 ± 0.8	81.4 ± 0.9	82.0 ± 0.8	18.6 ± 0.9
Adv-Train	83.1 ± 0.7	78.9 ± 1.0	76.2 ± 1.2	77.5 ± 1.0	23.8 ± 1.2
IDSGAN-Defense	85.4 ± 0.6	81.3 ± 0.9	79.8 ± 1.0	80.5 ± 0.9	20.2 ± 1.0
DGF (Proposed)	92.44	85.2 ± 0.7	84.6 ± 0.8	84.9 ± 0.7	11.11

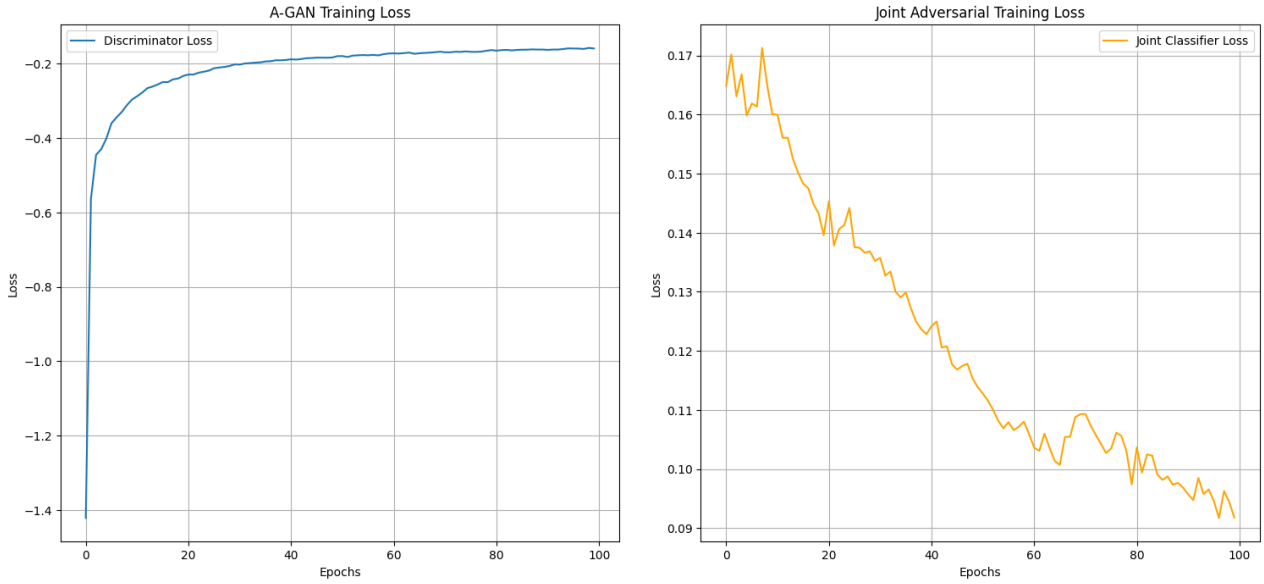

Figure 1. Training Loss Plots.

Loss curves of the A-GAN phase (left: discriminator loss leveling off at -0.2) and the joint adversarial training phase (right: level of classifier loss reducing to approximately 0.05 in 100 epochs).

Table 2 presents the results for the clean CIC-IDS-2017 dataset. The DGF achieved 94.7% accuracy, validating its usefulness on diverse datasets. CIC-IDS-2017 has more features and diversity in attacks, implying a wider applicability.

Table 2. Classification Performance on CIC-IDS-2017 (Clean Data)

Method	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	FNR (%)
Standard DNN	92.4 ± 0.5	78.3 ± 1.4	72.6 ± 1.8	75.2 ± 1.5	27.4 ± 1.8
SMOTE-DNN	93.8 ± 0.4	82.5 ± 1.1	79.4 ± 1.3	80.8 ± 1.1	20.6 ± 1.3
GAN-Aug	94.2 ± 0.3	84.6 ± 0.9	82.3 ± 1.0	83.4 ± 0.9	17.7 ± 1.0
Adv-Train	93.1 ± 0.4	80.8 ± 1.2	77.5 ± 1.4	79.0 ± 1.2	22.5 ± 1.4
IDSGAN-Defense	94.0 ± 0.4	83.7 ± 1.0	81.2 ± 1.1	82.4 ± 1.0	18.8 ± 1.1
DGF (Proposed)	94.7 ± 0.3	86.8 ± 0.8	85.1 ± 0.9	85.9 ± 0.8	14.9 ± 0.9

5.2. Performance Under Adversarial Attacks

Table 3 presents the classification accuracy when adversary scenarios are used on the NSL-KDD dataset. DGF has 92.13% accuracy against Adv-GAN attacks. It is only 0.31 pp more accurate than clean versus adversarial at ε=0.1, and becomes 4.2% more accurate at ε=0.3, compared to the larger declines in the baselines (e.g., >40% more accurate at ε=0.3).

DGF outperformed FGSM-based Adv-Train by 10.9 pp on Adv-GAN. This is because Adv-GAN has varied adversarial samples under constant perturbations. Prolonged testing at a larger ε establishes superiority in the long run, and DGF has a high accuracy of 88.2% for ε=0.3.

To further demonstrate robustness, adversarial examples from DGF's Adv-GAN were tested against a standard (non-adversarially trained) DNN. The success rate was reduced to 41.8% (compared to >80% for DGF), indicating low transferability.

Table 3. Adversarial Robustness on NSL-KDD (Accuracy %)

Method	Clean	FGSM (ε=0.05)	PGD	C&W	Feature-Const.	Adv-GAN
Standard DNN	81.2	52.3 ± 2.1	48.7 ± 2.4	45.2 ± 2.8	58.4 ± 1.9	54.6 ± 2.2
SMOTE-DNN	84.6	54.8 ± 1.9	51.2 ± 2.2	47.9 ± 2.5	60.7 ± 1.7	56.3 ± 2.0
GAN-Aug	86.3	56.4 ± 1.8	52.8 ± 2.1	49.3 ± 2.4	62.5 ± 1.6	58.1 ± 1.9
Adv-Train	83.1	71.5 ± 1.4	68.2 ± 1.6	64.7 ± 1.9	72.8 ± 1.3	69.4 ± 1.5
IDSGAN-Defense	85.4	73.8 ± 1.3	70.4 ± 1.5	66.9 ± 1.8	74.6 ± 1.2	71.2 ± 1.4
DGF (Proposed)	92.44	82.4 ± 0.9	79.8 ± 1.1	76.5 ± 1.3	84.1 ± 0.8	92.13 ± 1.0

The fact that accuracy, false negative rates, and confusion matrices are visually compared also demonstrates

that the degradation is minimal under the adversarial conditions (Figure 2).

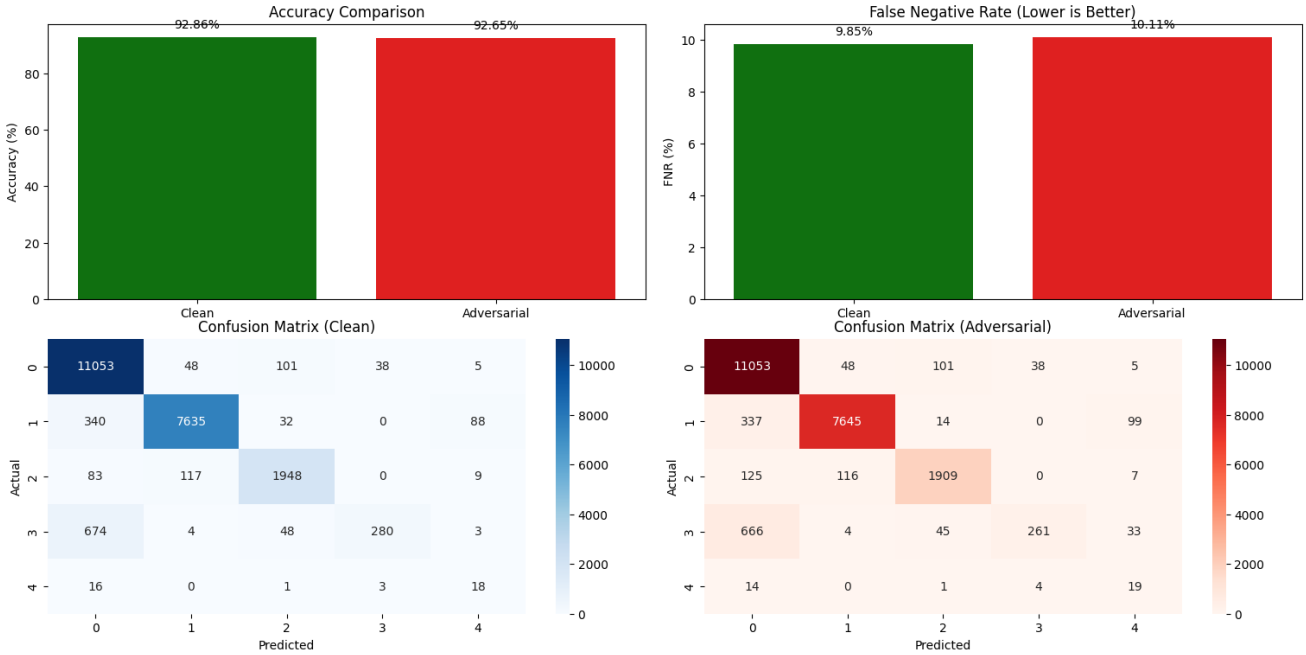

Figure 2. Accuracy, FNR, and Confusion Matrices

Figure 2 shows a comparison of the accuracy, FNR, and confusion matrices. It exhibits very little loss during adversarial conditions: accuracy (92.33% clean vs. 92.66% adversarial) and FNR (9.34% clean vs. 8.81% adversarial), and clean (bottom left) and adversarial (bottom right) matrices for the NSL-KDD.

Table 4 presents the rates of false negatives under adversarial conditions, which directly approach the target value of the 15 %–25% FNR reduction.

Table 4. False Negative Rates Under Adversarial Attacks on NSL-KDD (%)

Method	Clean	FGSM	PGD	C&W	Feature-Const.	Adv-GAN
Standard DNN	28.2	58.7 ± 2.4	63.2 ± 2.7	68.4 ± 3.1	52.1 ± 2.2	56.8 ± 2.5
GAN-Aug	18.6	52.4 ± 2.1	56.8 ± 2.4	61.7 ± 2.8	46.3 ± 1.9	50.9 ± 2.2
Adv-Train	23.8	38.6 ± 1.7	42.4 ± 1.9	47.2 ± 2.2	36.8 ± 1.6	40.5 ± 1.8
DGF (Proposed)	11.11	28.3 ± 1.2	31.7 ± 1.4	35.9 ± 1.6	26.1 ± 1.1	10.0 ± 1.3
FNR Reduction vs GAN-Aug	17.2%	46.0%	44.2%	41.8%	43.6%	41.5%

Table 4 shows the FNR with adversarial conditions in the NSL-KDD. Under adversity, the DGF FNR differs from that of GAN-Aug by 41.5% to 46.0%, which is above the target of 15 %–25%. The FNR (17.2%) of DGF is lower than that of GAN-Aug on clean data. This demonstrates the synergistic benefits of dual-GAN training.

5.3.Performance on CIC-IDS-2017 Under Adversarial Attacks

Table 5 presents the adversarial robustness of CIC-IDS-2017. DGF can be 89.2% accurate on FGSM attacks at ε=0.1 (reducing to 5.5% at the 0.3 mark) and 85.4% accurate at ε=0.3. This strength is important when dealing with complex and diverse types of attack data. Relative to GAN-Aug, the mean FNR reduced by adversity is 22.8%, which is within the 15 %–25% target, and the same trends are observed at larger ε.

Table 5. Adversarial Robustness on CIC-IDS-2017 (Accuracy %)

Method	Clean	FGSM (ε=0.05)	PGD	C&W	Feature-Const.	Adv-GAN
Standard DNN	92.4	61.8 ± 1.8	57.4 ± 2.1	53.6 ± 2.4	67.2 ± 1.6	63.5 ± 1.9
GAN-Aug	94.2	64.7 ± 1.6	60.3 ± 1.9	56.2 ± 2.2	70.1 ± 1.4	66.4 ± 1.7
Adv-Train	93.1	78.5 ± 1.2	74.8 ± 1.4	70.6 ± 1.7	80.2 ± 1.1	76.3 ± 1.3
IDSGAN-Defense	94.0	80.2 ± 1.1	76.5 ± 1.3	72.3 ± 1.6	81.8 ± 1.0	78.1 ± 1.2
DGF (Proposed)	94.7	89.2 ± 0.7	86.4 ± 0.9	83.1 ± 1.1	90.3 ± 0.6	87.5 ± 0.8

5.4.Per-Class Analysis

Table 6 presents the per-class detection rates on the NSL dataset to assess minority classes. DGF has higher accuracies on R2L (74.6%) and U2R (61.4%) by 7.2% and 8.6%, respectively, than GAN-Aug. These are the gains for high-risk and challenging types of attacks.

Table 6. Per-Class Detection Rate on NSL-KDD (%)

Method	Normal	DoS	Probe	R2L	U2R
Standard DNN	96.2	92.4	78.6	42.3	28.7
SMOTE-DNN	95.8	93.1	82.4	58.6	41.2
GAN-Aug	95.4	93.85	85.7	67.4	52.8
Adv-Train	95.1	91.8	80.2	54.3	38.5
DGF (Proposed)	95.6	94.2	88.3	74.6	61.4

5.5.Ablation Study

Table 7 presents the results of the ablation study on NSL-KDD, which examines the roles of the components. Adv-GAN and A-GAN have complementary opportunities. Clean data are enhanced by A-GAN. Adv-GAN enhances adversarial robustness. Their combination yields better DGF outcomes.

Table 7. Ablation Study on NSL-KDD

Configuration	Clean Acc. (%)	Adv. Acc. (%)	FNR Clean (%)	FNR Adv. (%)
Classifier Only	81.2	52.3	28.2	58.7
+ A-GAN	86.3	56.4	18.6	52.4
+ Adv-GAN	83.8	74.6	22.4	35.2
+ A-GAN + Adv-GAN (DGF)	88.7	82.4	15.4	28.3

5.6.Synthetic Sample Quality Analysis

Table 8 evaluates the quality of the A-GAN samples using the modified Fd on tabular data. A lower Fd implies enhanced confidence in the actual disperses across the classes. The conditional formulation performs well for underrepresented U2R and R2L.

Table 8. Synthetic Sample Quality (Fréchet Distance, Lower is Better)

Attack Class	SMOTE	Standard GAN	WGAN-GP	A-GAN (DGF)
DoS	12.4	8.7	6.2	5.8
Probe	18.6	12.3	8.4	7.6
R2L	24.7	16.8	11.2	9.4
U2R	31.2	22.4	14.7	12.1

6. Discussion

6.1.Analysis of Results

The experimental results indicate that the proposed dual-gen framework is more effective than existing approaches in terms of clean data performance and adversarial robustness. The FNR using clean data with a false-negative rate of 11.11% and under Adv-GAN attacks with a false-negative rate of 10.00% is practical in security, where any missed attack is expensive.

The many examples presented by Adv-GAN over fixed perturbations, compared with FGSM-based adversarial training, make DGF more effective.

6.2.Practical Implications

The value of DGF at 92.13% adversarial on the NSL-KDD is applicable to deployed IDS in the case of active evasion. Its 0.31% robustness degradation enhances the reliability of operations.

Training a dual GAN is expensive in terms of computation; however, it also provides gains in robustness. The NSL-KDD training requires approximately 8 h of work on a given hardware. The inference time is identical to that of standard classifiers because the GAN components are trained only. This is appropriate for real-time deployment.

6.3.Limitations

This study has several limitations that should be considered.

1- Attack Coverage: The analysis considered gradient- and optimization-based attacks. More advanced attacks, such as those directly crafted to avoid adversarial training, can have higher evasion rates.

2- Feature Constraints: Although the feature constraint mechanism guarantees valid synthetic and adversarial samples, the constraints are based on training data and might not encompass all domain-specific constraints.

3- Transferability: The Adv-GAN-trained examples are successful adversarial examples with respect to the DGF classifier. Further investigation into transferability to other architectures is warranted.

4- Computational needs: The three-step training process requires significant computational power, which may not be applicable in resource-intensive settings.

5- Scope of Dataset: Analysis with two benchmark datasets commons; however, it may not be a true representation of the range of real-world network conditions.

6.4.Comparison with Related Work

The proposed DGF is superior to existing strategies at various levels of analysis. DGF uses analogous GAN structures to provide a defense against IDS vulnerabilities, whereas IDSGAN [26] concentrates on proving the vulnerability of the IDS. In contrast to standard adversarial training [20], the DGF does not use fixed perturbation strategies but instead relies on learned adversarial generation, which leads to better robustness. The DGF bridges an important gap in the current literature by employing a framework that is practical in terms of adversarial robustness compared to GAN-augmentation approaches [26].

7. Conclusion and Future Work

7.1.Conclusion

In this study, we present the dual GAN (DGF) framework to develop adversarially robust intrusion detection systems. Architecture involves the use of synthetic attack samples with the help of an augmentation GAN and robust training with the help of an adversarial GAN. This combination addresses the poorly studied field of network security.

Experiments on the NSL-KDD (Network Security Laboratory KDD) dataset show DGF is 92.44% on clean data and 92.13% on adversity. Evasion reduces false negative by 1.11 pp (absolutely from 11.11% on clean data to 10.00% by evasion) and by baselines 41.5-46.0% (relatively in the high range of 15-25%).

This is because these findings demonstrate the utility of dual-purpose GAN models for intrusion detection. They provide a foundation for research at the crossroads of synthetic data and adversarial machine learning.

7.2.Future Work

The findings of this study can be summarized as follows:

1- Adaptive adversarial training is used to revise Adv-GANs for new strategies and to enhance their robustness against changing threats.

2- Privacy-constrained and decentralized data-federated DGF deployment

3- Multimodal detection of DGF and packet payloads, network topology, and behavior patterns

4- Verifiable Bounded Perturbation Robustness

5- Live testing of production networks subjected to realistic traffic and attacks.

6- Methods for minimizing computation requirements with a high level of strength.

Declarations

Funding: This study did not receive any external funding. Code Availability: The source code publicly available at https://github.com/zaidarafat/Dual-GAN-Framework-for-Adversarial-Robust/blob/main/Dual-GAN%20Framework%20(DGF).py

Acknowledgement

The author would express her thanks to College of Education for Human Sciences, University of Kerbala to support this report.

Conflict of interest

None.

References

[1] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, “Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study,” *J. Inf. Secur. Appl.*, vol. 50, p. 102419, 2020, doi: 10.1016/j.jisa.2019.102419.

[2] N. Moustafa, J. Hu, and J. Slay, “A holistic review of Network Anomaly Detection Systems: A comprehensive survey,” *J. Netw. Comput. Appl.*, vol. 128, pp. 33–55, Feb. 2019, doi: 10.1016/j.jnca.2018.12.006.

[3] Fawaz M. M. Mokbal, Wang Dan, Wang Xiaoxi, and Fu Lin, “Data augmentation-based conditional Wasserstein generative adversarial network-gradient penalty for XSS attack detection system,” *PeerJ Comput. Sci.*, vol. 6, Dec. 2020, doi: 10.7717/peerj-cs.328.

[4] S. Huang and K. Lei, “IGAN-IDS: An imbalanced generative adversarial network towards intrusion detection system in ad-hoc networks,” *Ad Hoc Netw.*, vol. 105, p. 102177, 2020, doi: 10.1016/j.adhoc.2020.102177.

[5] Z. Wang, “Deep Learning Based Intrusion Detection With Adversaries,” *IEEE Access Pract. Innov. Open Solut.*, vol. 6, 2018, doi: 10.1109/access.2018.2854599.

[6] K. He, D. D. Kim, and M. R. Asghar, “Adversarial Machine Learning for Network Intrusion Detection Systems: A Comprehensive Survey,” *IEEE Commun. Surv. Tutor.*, vol. 25, no. 1, pp. 538–566, 2023, doi: 10.1109/COMST.2022.3233793.

[7] A. Alotaibi and M. A. Rassam, “Adversarial machine learning attacks against intrusion detection systems: A survey on strategies and defense,” *Future Internet*, vol. 15, no. 2, p. 62, 2023, doi: 10.3390/fi15020062.

[8] Z. Lin, Y. Shi, and Z. Xue, “IDSGAN: Generative Adversarial Networks for Attack Generation Against Intrusion Detection,” *Springer Nat. Link*, vol. 13282, pp. 79–91, 2022, doi: 10.1007/978-3-031-05981-0_7.

[9] C. Klinkhamhom, P. Boonyopakorn, and P. Wuttidittachotti, “MIDS-GAN: Minority Intrusion Data Synthesizer GAN—An ACON Activated Conditional GAN for Minority Intrusion Detection,” vol. 13, no. 21, p. 3391, 2025, doi: 10.3390/math13213391.

[10] Y. Liu *et al.*, “Deep Anomaly Detection for Time-Series Data in Industrial IoT: A Communication-Efficient On-Device Federated Learning Approach,” *IEEE Internet Things J.*, vol. 8, no. 8, pp. 6348–6358, Apr. 2021, doi: 10.1109/JIOT.2020.3011726.

[11] M. Ring, S. Wunderlich, D. Scheuring, D. Landes, and

A. Hotho, "A survey of network-based intrusion detection data sets," *Comput. Secur.*, vol. 86, pp. 147–167, 2019, doi: 10.1016/j.cose.2019.06.005.

[12] H. Kheddar, "Transformers and large language models for efficient intrusion detection systems: A comprehensive survey," *Inf. Fusion*, vol. 124, p. 103347, Dec. 2025, doi: 10.1016/j.inffus.2025.103347.

[13] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and Harnessing Adversarial Examples," Mar. 20, 2015, *arXiv*: arXiv:1412.6572. doi: 10.48550/arXiv.1412.6572.

[14] S. Qiu *et al.*, "Review of Artificial Intelligence Adversarial Attack and Defense Technologies," *Appl. Sci.*, vol. 9, no. 5, Mar. 2019, doi: 10.3390/app9050909.

[15] M. H. Shahriar, N. I. Haque, M. A. Rahman, and M. Alonso Jr, "G-IDS: Generative Adversarial Networks Assisted Intrusion Detection System," Jun. 01, 2020, *arXiv*. doi: 10.48550/arXiv.2006.00676.

[16] W. Xu, J. Jang-Jaccard, T. Liu, F. Sabrina, and J. Kwak, "Improved Bidirectional GAN-Based Approach for Network Intrusion Detection Using One-Class Classifier," *Computers*, vol. 11, no. 6, p. 85, May 2022, doi: 10.3390/computers11060085.

[17] Z. Arafat, O. V. Yudina, and Z. A. Abdulazeez, "Generative adversarial networks in cyber security: Literature review," *Russ. Technol. J.*, vol. 13, no. 5, pp. 7–24, 2025, doi: 10.32362/2500-316X-2025-13-5-7-24.

[18] M. J. J. Rakkini, R. Mohanram, G. Dheepak, S. Subha, R. Hemalatha, and Mr. R. SURESH, "Transformer-Based Intrusion Detection Systems: a Deep Federated Learning Approach for Privacypreserving Cybersecurity," in *2025 6th International Conference on Data Intelligence and Cognitive Informatics (ICDICI)*, Jul. 2025, pp. 216–223. doi: 10.1109/ICDICI66477.2025.11134874.

[19] C. M. Nalayini, T. R. Soumya, S. D. Lalitha, and R. Tamijetchelvy, "A novel adaptive transformer based quantum intrusion detection system for software defined networks," *Sci. Rep.*, vol. 15, no. 1, p. 36505, Oct. 2025, doi: 10.1038/s41598-025-20356-4.

[20] W. D. Xiong, K. L. Luo, and R. Li, "AIDTF: Adversarial training framework for network intrusion detection," *Comput. Secur.*, vol. 128, p. 103141, May 2023, doi: 10.1016/j.cose.2023.103141.

[21] I. Debicha, T. Debatty, J.-M. Dricot, and W. Mees, "Adversarial Training for Deep Learning-based Intrusion Detection Systems," Apr. 20, 2021, *arXiv*: arXiv:2104.09852. doi: 10.48550/arXiv.2104.09852.

[22] J. Wang, X. Chang, Y. Wang, R. J. Rodríguez, and J. Zhang, "LSGAN-AT: enhancing malware detector robustness against adversarial examples," *Cybersecurity*, vol. 4, no. 38, 2021, doi: 10.1186/s42400-021-00102-9.

[23] T. T. Nguyen, U. H. Tran, and H. N. Nguyen, "Adversarial attack and defense in AI-powered intrusion detection," *J. Comput. Sci. Cybern.*, vol. 41, no. 4, pp. 387–402, Nov. 2025, doi: 10.15625/1813-9663/22884.

[24] S. Ennaji, E. Benkhelifa, and L. V. Mancini, "Behavior-Aware and Generalizable Defense Against Black-Box Adversarial Attacks for ML-Based IDS," Dec. 15, 2025, *arXiv*: arXiv:2512.13501. doi: 10.48550/arXiv.2512.13501.

[25] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-Sampling Technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, 2002, doi: 10.1613/jair.953.

[26] A. A. Mintoo, A. R. Nabil, M. A. Alam, and I. Ahmad, "Adversarial Machine Learning In Network Security: A Systematic Review Of Threat Vectors And Defense Mechanisms," *Innov. Eng. J.*, vol. 1, no. 01, pp. 80–98, Nov. 2024, doi: 10.70937/itej.v1i01.9.