mathematics

MDPI

Article

MIDS-GAN: Minority Intrusion Data Synthesizer GAN—An ACON Activated Conditional GAN for Minority Intrusion Detection

Chalerm Klinkhamhom, Pongsarun Boonyopakorn * and Pongpisit Wuttidittachotti

Department of Network and Information Security Management, King Mongkut's University of Technology North Bangkok, Bangkok 10800, Thailand; s6407031910035@kmutnb.ac.th (C.K.); pongpisit.w@itd.kmutnb.ac.th (P.W.)
* Correspondence: pongsarun.b@itd.kmutnb.ac.th

Abstract

Intrusion Detection Systems (IDS) are vital to cybersecurity but suffer from severe class imbalance in benchmark datasets such as NSL-KDD and UNSW-NB15. Conventional oversampling methods (e.g., SMOTE, ADASYN) are efficient yet fail to preserve the latent semantics of rare attack behaviors. This study introduces the Minority-class Intrusion Detection Synthesizer GAN (MIDS-GAN), a divergence-minimization framework for minority data augmentation under structured feature constraints. MIDS-GAN integrates (i) correlation-based structured feature selection (SFS) to reduce redundancy, (ii) trainable ACON activations to enhance generator expressiveness, and (iii) KL-divergence-guided alignment to ensure distributional fidelity. Experiments on NSL-KDD and UNSW-NB15 demonstrate significant improvement on detection, with recall increasing from 2% to 27% for R2L and 1% to 17% for U2R in NSL-KDD, and from 18% to 44% for Worms and 69% to 75% for Shellcode in UNSW-NB15. Weighted F1-scores also improved to 78%, highlighting MIDS-GAN's effectiveness in enhancing minority-class detection through a principled, divergence-aware approach.

Keywords: intrusion detection systems; cybersecurity; Generative Adversarial Networks; data balancing; minority-class synthesis

MSC: 68U35

check for updates

Academic Editor: Jiang Guo

Received: 23 September 2025
Revised: 19 October 2025
Accepted: 22 October 2025
Published: 24 October 2025

Citation: Klinkhamhom, C.; Boonyopakorn, P.; Wuttidittachotti, P. MIDS-GAN: Minority Intrusion Data Synthesizer GAN—An ACON Activated Conditional GAN for Minority Intrusion Detection. *Mathematics* **2025**, *13*, 3391. https://doi.org/10.3390/math13213391

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Modern digital infrastructures, including cloud-native services, enterprise IoT, and 5G or edge environments, are becoming increasingly connected and complex, which in turn raises the risk of security breaches and cyberattacks. Network Intrusion Detection Systems (IDSs) are vital for cybersecurity, but their effectiveness depends on training data that capture both common and rare attack behaviors [1]. Widely used benchmarks such as NSL-KDD, UNSW-NB15, and CICIDS suffer from severe class imbalance, causing models to focus on frequent attacks while overlooking rare yet important intrusions, which remains a major limitation in IDS performance [2].

A range of studies have examined datasets and methodologies to address these challenges. Early studies emphasized the need for reliable datasets, whereas subsequent ones identified their persistent imbalance and outdated nature. Recent reviews highlight the potential of Generative Adversarial Networks (GANs) to mitigate class imbalance; however,

broader analyses continue to report enduring issues of dataset bias, limited generalizations, and uneven attack representation [3–5].

GANs offer a distribution-learning alternative to traditional oversampling by modelling minority-class manifolds through adversarial training to generate more diverse and realistic synthetic data. Variants such as Conditional GAN (CGAN), Wasserstein GAN with Gradient Penalty (WGAN-GP), and imbalance-aware GANs have shown significant improvements in recall and F1-scores for minority classes [6]. Specifically, CGAN employs label conditioning to guide data synthesis, while Auxiliary Classifier Generative Adversarial Network (ACGAN) applies a secondary classifier to enhance class consistency, ensuring that generated samples remain realistic and accurate—crucial for maintaining IDS reliability.

Activation functions critically influence IDS performance by defining decision boundaries. Although Rectified Linear Unit (ReLU) remains dominant, its rigid thresholding and "dying neuron" issues hinder the modelling of sparse or irregular minority regions [7]. Recent studies show that trainable activation enhances representational capacity and gradient stability. Accordingly, the Adaptive Concatenated Activation (ACON) [8] is adopted—a learnable function that dynamically interpolates between linear and nonlinear behaviors via gating. ACON mitigates gradient vanishing and enhances feature expressiveness, enabling both discriminator and generator to better capture ultra-minority classes.

To address the gap in achieving class-conditional generation with consistent labelling and stable distributional alignment in IDS data synthesis, this study proposes the Minority Intrusion Detection Synthesizer GAN (MIDS-GAN), which integrates an ACGAN backbone, the adaptive ACON activation function, and KL-divergence–guided training. The functions of these core components are as follows:

- Class-conditional generation with ACGAN to produce label-consistent minority samples.
- Trainable activation (ACON) to adaptively model complex, imbalanced intrusion manifolds.
- Structured feature selection with KL-divergence alignment to preserve distributional fidelity and mitigate mode dropping.

Unlike traditional oversampling, MIDS-GAN generates synthetic traffic that is both statistically consistent and semantically aligned with real intrusions. Results presented in Section 4 demonstrate significant improvements in recall and F1-scores for minority classes while maintaining distributional integrity. By integrating cybersecurity, generative modelling, and imbalance-aware learning, MIDS-GAN offers a principled framework for more effective and interpretable IDS development, enhancing the detection of minority intrusions.

This paper comprises six sections. Section 2 reviews related work and datasets, identifying the critical research gap that motivates this study. Section 3 details the proposed MIDS-GAN methodology, while Section 4 describes the experimental design and evaluation settings. Sections 5 and 6 present the discussion and conclusion, respectively.

2. Related Work

2.1. Oversampling Techniques

Classical oversampling is a common approach to addressing class imbalance. Methods such as the Synthetic Minority Oversampling Techniques (SMOTE) [9], ADASYN [10], and their hybrids (e.g., SMOTE-ENN [11] and Borderline-SMOTE [12]) are widely used in IDS research for their computational efficiency. These geometry-based techniques generate synthetic minority samples by interpolating between nearest neighbors but suffer from redundancy, oversmoothing, class overlap, and overfitting risks [13]. Adaptive variants, such as the Adaptive SV-Borderline SMOTE-SVM proposed by Guo et al. [14], aim to refine decision boundaries, while regression-specific and quantum-inspired extensions broaden their applicability. Nonetheless, oversampling remains limited to feature-space heuristics

and fails to capture the latent semantics of cyberattack behaviors, highlighting the need for distribution-learning approaches such as generative models.

2.2. Generative Adversarial Networks (GANs) for Intrusion Detection

GANs offer a distinct paradigm by learning the underlying data distribution through adversarial training between a generator and discriminator, rather than interpolating in feature space [15]. This makes them well-suited for synthesizing minority-class traffic that remains semantically consistent with real attacks. Early applications in IDS showed promising results. Yin et al. [16] applied recurrent neural networks (RNNs) to model sequential traffic, resulting in improved detection on benchmark datasets. Yang et al. [17] combined a Deep Convolutional GAN (DCGAN) with a Simple Recurrent Unit (SRU), demonstrating that adversarial and temporal modeling together enhance classification performance on KDD'99 and NSL-KDD. Accordingly, Shahriar et al. [18] proposed G-IDS, where a GAN synthesizes minority samples and jointly trains with real and synthetic data, yielding greater detection stability on NSL-KDD.

To address class imbalance, Dlamini and Fahim [19] proposed the Data Generative Model (DGM), a conditional GAN leveraging KL-divergence to enhance distributional fidelity. Their results on NSL-KDD and UNSW-NB15 showed that distribution-aware synthesis improve detection of ultra-minority classes compared with heuristic oversampling methods such as SMOTE and ADASYN. Later studies identified additional challenges in adversarial settings, noting that carefully crafted perturbations can evade conventional IDS, while adversarial training can improve robustness [20]. Based on these findings, our recent study [21] demonstrated that conditional variants, particularly CGAN and ACGAN outperform vanilla GANs in detecting minority classes on NSL-KDD. Accordingly, this present work integrates adaptive ACON activations and structured feature selection into an ACGAN backbone. The resulting MIDS-GAN framework extends DGM principle, explicitly enforcing label fidelity and distributional alignment to enhance synthesis and detection of ultra-minority intrusions in NSL-KDD and UNSW-NB15.

Recent architectures such as WGAN-GP [22], WGAN-DL-IDS [23], and the DRL-GAN hybrid model [24] have achieved notable improvements in detection accuracy on complex datasets like UNSW-NB15 and in IoT network environments. However, these models often result in high computational complexity and limited generalization across heterogeneous datasets. Their performance also depends heavily on activation functions, which influence gradient stability and representational capacity. Recent surveys indicate that trainable activations offer greater adaptability than fixed functions such as ReLU, which potentially improves the quality of synthetic intrusion samples.

More advanced conditional variants have been developed to address dataset imbalance. Conditional GANs (CGANs), first introduced by Mirza and Osindero [25], integrates class labels into both the generator and discriminator to guide minority-class synthesis, but do not explicitly enforce label fidelity. Odena et al. [26] extended this approach with the ACGAN, which adds an auxiliary classifier to the discriminator, introducing a dual objective: adversarial loss for realism and classification loss for label consistency.

In IDS applications, such dual objectives are crucial, as mislabeled synthetic intrusions can degrade detection performance more severely than class imbalance itself. Variants such as AE-CGAN and other distribution-alignment methods emphasize feature compression or KL-divergence minimization but often underutilize ACGAN's classifier advantages. Recent studies on ACGAN-assisted IDS [27] confirm its effectiveness in improving minority-class detection. Yang et al. [28] further advanced this approach with SPE-ACGAN, which incorporates a Sample Partition and Enhancement (SPE) resampling strategy into the ACGAN framework, demonstrating superior performance on benchmark IDS datasets

compared with conventional oversampling methods. This progression from CGAN to ACGAN and its recent extensions thus motivates the adoption of ACGAN backbone in this study, whose auxiliary classifier ensures label fidelity essential for synthesizing ultra-minority attack categories.

2.3. Summary and Motivation

GAN–based intrusion detection systems have shown strong potential for mitigating severe class imbalance. Prior studies [19,22,27] report measurable improvements in recall, F1-score, and overall detection accuracy on benchmark datasets such as NSL-KDD and UNSW-NB15. Collectively, these studies highlight three key insights.

(1) Effectiveness of adversarial generation—GANs can synthesize minority-class traffic patterns that traditional oversampling fails to capture, thereby improving minority detection rates.
(2) Technical challenges—Training instability, mode collapse, and the absence of strict label fidelity remain significant obstacles, especially for ultra-minority classes (e.g., R2L, U2R).
(3) Limitations of conventional oversampling—Although computational efficient, methods such as SMOTE and ADASYN rely on geometric heuristics and cfail to preserve the behavioral semantics of cyberattacks.

Accordingly, this present study identifies a key research gap: the need for class-conditional generation with guaranteed label consistency and stable distributional alignment. ACGANs offer a foundation for label-faithful synthesis but require adaptive activation and explicit distribution matching to achieve optimal performance. To address these challenges, the proposed MIDS-GAN framework integrates (i) an ACGAN backbone for conditional generation, (ii) the adaptive ACON activation function to enhance representational flexibility, and (iii) KL-divergence–guided training to enforce distributional stability. This principled design directly addresses the limitations of prior IDS augmentation methods and provides a robust framework for improving minority intrusion detection.

3. Proposed Methodology

This section outlines the augmentation module, feature selection process, and overall architecture of the proposed MIDS-GAN.

3.1. Overview of the MIDS-GAN Augmentation Module (Train-Time Only)

The core augmentation module of MIDS-GAN integrates three mechanisms: (i) conditional generation via ACGAN for label-faithful synthesis, (ii) class-wise KL-divergence regularization for distributional alignment, and (iii) trainable ACON activation to enhance nonlinearity and stability in sparse minority regions. It functions exclusively during training (Figure 1a). Given minority samples $x \mid y$ and their labels y, the generator $G(z \mid y)$ (ACON) synthesizes class-conditional flows $\widetilde{x} \mid y$ from random noise z. The discriminator D jointly learns adversarial and auxiliary classification objectives through $\mathcal{L}_{adv}$ and $\mathcal{L}_{cls}$, while a class-wise divergence penalty ($\lambda_{KL}\mathcal{L}_{KL}$) aligns real and generated feature distributions to enforce statistical fidelity, guiding the generator toward more faithful minority manifolds. The resulting synthetic data are combined with real samples to form an augmented training set, whereas testing always relies on real data only to prevent bias leakage.

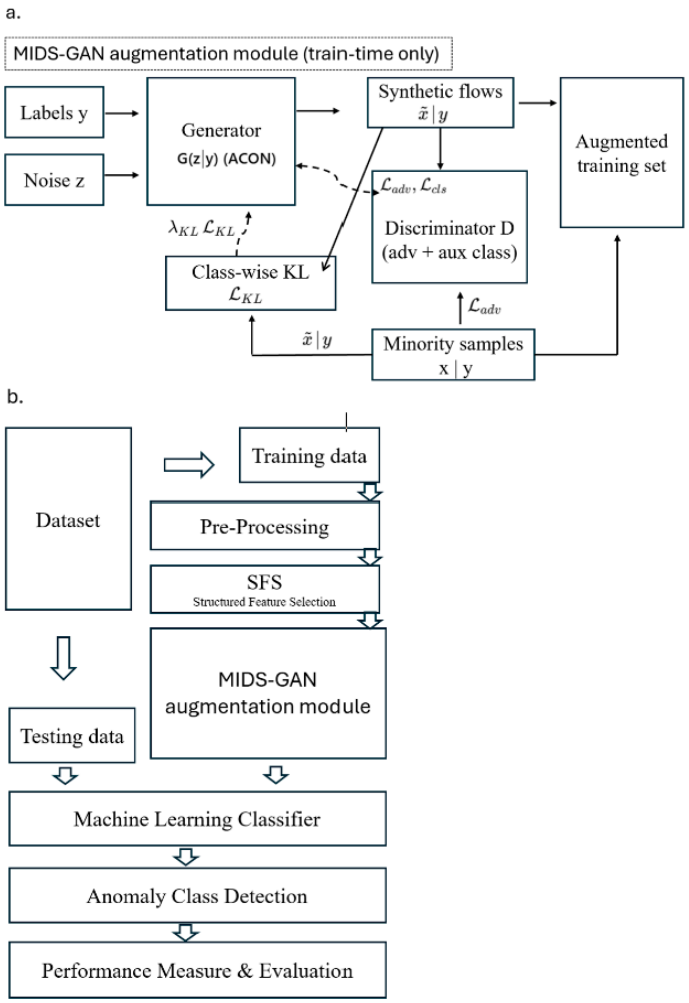

Figure 1. Overview of the proposed MIDS-GAN framework: (**a**) train-time augmentation module using ACON-activated ACGAN with class-wise KL alignment; and (**b**) integration into the end-to-end IDS workflow encompassing preprocessing, structured feature selection, and classifier evaluation.

Figure 1b shows the integration of the MIDS-GAN augmentation module within the IDS workflow, where data preprocessing and structured feature selection (SFS) precede the augmentation, followed by classifiers training.

The anomaly detection task is formulated as a multi-class supervised learning problem. Let $X \in R^{n \times d}$ denote the training dataset with n samples and d features, and $Y = \{y_1, y_2, \ldots, y_n\}$ denote the corresponding class labels. The objective is to learn a function $f : X \rightarrow Y$ that generalizes effectively to unseen data, with particular emphasis on underrepresented (minority) classes.

3.2. Feature-Driven Preprocessing

Before model development, the raw dataset undergoes a systematic preprocessing pipeline to ensure consistency, numerical compatibility, and scale normalization—which is crucial for neural network-based models like GANs. This process includes the following steps:

3.2.1. Encoding of Categorical Features

Network intrusion datasets contain both numerical and categorical variables, requiring categorical features to be transformed into numeric representations to ensure neural network compatibility. Early approaches to encoding categorical features relied on simple methods such as one-hot or ordinal, while more recent studies have adopted deep-learned embedding

techniques [29]. However, naive encodings can distort joint distributions between categorical and continuous variables, thereby reducing generative fidelity which is a limitation frequently emphasized in tabular learning literature [30]. To overcome this, type-aware models such as Conditional Tabular GAN (CTGAN) [31] and CTAB-GAN+ [32] employ specialized encoders and mode-specific normalization to preserve categorical–continuous dependencies. This capability is crucial in IDS applications, where protocol types, service categories, and flag attributes must accurately interact with numerical traffic statistics.

3.2.2. Normalization and Scaling

For numerical features, a two-step normalization strategy is applied, beginning with unit-norm scaling followed by robust scaling based on the interquartile range (IQR). This approach mitigates the impact of outliers and heavy-tailed distributions common in intrusion detection datasets (e.g., burst traffic or skewed packet size). Recent generative modelling studies [32,33] further recommend mode-specific normalization to accommodate multimodal continuous attributes, providing a more expressive basis for synthesizing realistic intrusion traffic samples.

3.2.3. Constant Feature Removal

Features exhibiting extremely low variance (i.e., constant in more than 99.5% of samples) are removed, as they contribute little to learning and may increase overfitting or unnecessary parameter complexity. The resulting dataset from this preprocessing pipeline offers a cleaner and more numerically stable foundation for subsequent feature selection and model training.

3.3. Structured Feature Selection: Correlation-Based Filtering (SFS)

To further improve the quality of the input feature space, a structured feature selection step based on correlation analysis is applied. This process identifies and remove highly correlated features that may cause multicollinearity, redundancy, and overfitting—particularly problematic for deep generative models such as GANs.

For each dataset, a pairwise correlation matrix is computed among continuous features. The linear relationship between two features, X and Y, is quantified using the Pearson correlation coefficient [34], defined as

$$\rho_{X,Y} = \frac{\sum_{i=1}^{n}(X_i - \overline{X})(Y_i - \overline{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \overline{X})^2} \cdot \sqrt{\sum_{i=1}^{n}(Y_i - \overline{Y})^2}} \quad (1)$$

where

X_i *and* Y_i represent the values of features X and Y for sample i.

$\overline{X}$ and $\overline{Y}$ represent the means of X and Y, respectively.

n represents the number of data samples.

Features with *Threshold* : $|\rho| > \theta$ (where θ is a predefined threshold, e.g., 0.95) are considered highly correlated. Among these feature pairs, one is removed based on heuristic criteria such as higher mean absolute correlation with other features or lower relevance to minority classes.

In this study, a conservative upper bound of $\theta = 0.95$ is adopted to ensure strong decorrelation and minimize multicollinearity among intrusion detection features. This option aligns with prior study, where Khan et al. [35] used a 95% correlation cutoff for redundancy removal in Smart City intrusion detection, and Sabilla et al. [36] identified $r \geq 0.95$ as the optimal correlation threshold for feature reduction in correlation-based selection. Similar thresholds are also recommended in correlation-filtering studies to maintain sufficient feature independence in high-dimensional IDS datasets.

This approach is inspired by Correlation-Based Feature Selection (CFS) [37] and has been widely applied in domains such as text classification [38] and intrusion detection [39]. More recently, hybrid feature selection strategies combining correlation measures with wrapper-based refinement have been proposed for network intrusion detection, highlighting the continued relevance of correlation-aware methods in cybersecurity [40].

3.4. Conditional Generation via ACGAN with ACON Activation

To generate synthetic samples for specific intrusion classes, a conditional generative adversarial network (cGAN) is employed in the form of an ACGAN [26], integrated with ACON [8]. This configuration enables the generator to produce class-specific outputs, while the discriminator simultaneously evaluates sample authenticity and class label consistency. The internal architecture of the proposed model is illustrated in Figure 2, highlighting the interaction between the generator and the dual-headed discriminator as described below.

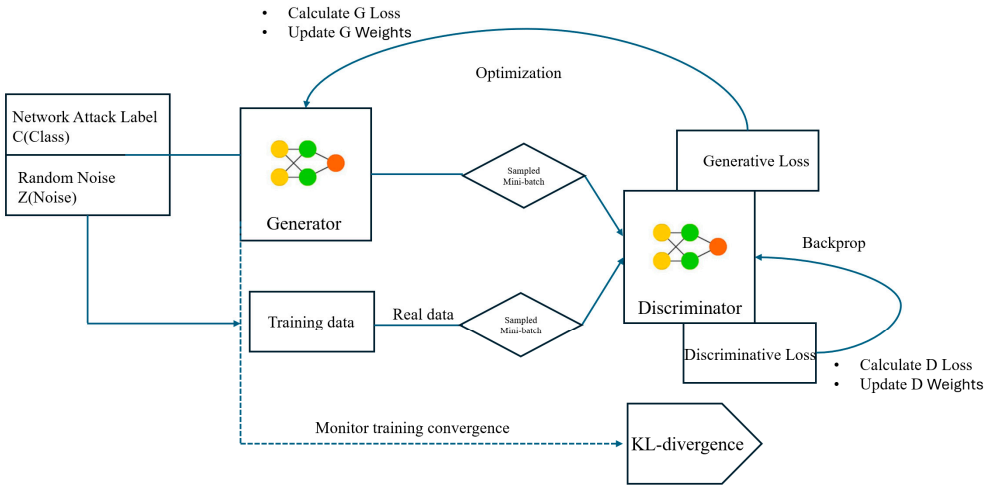

Figure 2. Architecture of the proposed ACGAN integrated with ACON. The generator produces class-conditional samples, while the discriminator jointly evaluates authenticity (real or fake) and class label using a dual-output structure.

The generator receives a noise vector $z \sim \mathcal{N}(0, I)$ concatenated with a class label $c \in \mathcal{C}$, enabling the synthesis of class-conditioned samples. The generated output is expressed as

$$\tilde{x} = G(z, c) \tag{2}$$

The discriminator has a dual-head structure. Its outputs include $D_{source}(x) \in [0, 1]$, representing the probability that x is real, and $D_{class}(x) \in R^{|C|}$, a softmax distribution over class labels. This architecture enables the model to learn both data realism and class consistency simultaneously.

The architectural design supports the dual-loss function in ACGAN: the source loss (L_{source}) promotes realism, while the class loss (L_{class}) ensures label-consistent in synthetic samples. By integrating ACON [8], both generator and discriminator benefit from trainable, adaptive activation dynamics that enhance gradient flow and representation power. The interaction between these objectives, combined with adaptive activations, is crucial for generating high-fidelity data for minority intrusion classes.

The generator mapping function is defined over latent and conditional spaces as follows:

$$z \in R^k,\ c \in \{1, \ldots, C\},\ G : R^k \times C \to R^d \tag{3}$$

The output of the generator is

$$G(z, c) = x \in R^d \tag{4}$$

The discriminator is modeled as a joint function that outputs both the real/fake authenticity score and the corresponding class prediction:

$$D(x) = (D_{\text{source}}(x),\ D_{\text{class}}(x)) \tag{5}$$

The class prediction is calculated using the softmax activation over $|\mathcal{C}|$ classes:

$$D_{class,i}(x) = \frac{exp(w_i^T x)}{\sum_{j=1}^{|C|} exp\left(w_j^T x\right)},\ i = 1, \ldots, |C| \tag{6}$$

To assess training convergence, the problem is formulated as a divergence minimization task. Given the real distribution $P_r(x \mid c)$ and the generator-induced distribution $P_g(x \mid c; \theta_G)$, the class-conditional KL divergence [26,33,41] is evaluated under structured feature constraints $\mathcal{S}$, ensuring that generated samples remain statistically aligned with their real counterparts

$$\mathrm{KL}\left(\mathrm{P_r}(.|\mathrm{c}) \parallel \mathrm{P_g}(.|\mathrm{c};\ \theta_\mathrm{G})\right) = \mathrm{E}_{\mathrm{x}\sim\mathrm{P_r}(.|\mathrm{c})}\left[\log\left(\mathrm{P_r}(\mathrm{x}|\mathrm{c})\ /\ \mathrm{P_g}(\mathrm{x}|\mathrm{c};\ \theta_\mathrm{G})\right)\right] \tag{7}$$

Here, θ_G denotes the generator parameters, and a lower KL value indicates that $P_g(x \mid c; \theta_G)$ more closely approximates $P_r(x \mid c)$.

3.4.1. Full Loss Function Formulation

To ensure consistency, both the discriminator and generator losses are formulated in a minimization form using negative log-likelihood for the source and class terms, consistent with ACGAN framework [26].

Discriminator loss:

$$L_D = E_{(x,c)\sim p_{data}}[-log\ D_{source}(x) - \lambda_{cls}\ log\ P(c|x)] + E_{(z,c)\sim p_{(z,c)}}[-log(1 - D_{source}(G(z,c)))] \tag{8}$$

Generator loss:

$$\mathrm{L_G} = \mathrm{E}_{(\mathrm{z,c})\sim\mathrm{p}_{(\mathrm{z,c})}}[-\log \mathrm{D_s ource}(\mathrm{G}(\mathrm{z},\mathrm{c})) - \lambda_\mathrm{c}\mathrm{ls}\ \log \mathrm{P}(\mathrm{c}|\mathrm{G}(\mathrm{z},\mathrm{c}))] \tag{9}$$

Note: In the original ACGAN, the class-loss term can be applied to real samples, fake samples, or both. In this study, the simplified and widely adopted formulation is used, where the class term is applied as shown above, ensuring full consistency with the minimization objective.

3.4.2. Trainable Activation: ACON

To enhance the representational capacity of both the generator and discriminator, traditional ReLU activations are replaced with the Adaptive Concatenated Activation (ACON) function, defined as

$$ACON(x) = (p_1 - p_2)x \cdot \sigma[\beta(p_1 - p_2)x] + p_2 x \tag{10}$$

where

$\sigma(\cdot)$ denotes the sigmoid function.

p_1 *and* p_2 are learnable parameters controlling the linear bounds.

β is a learnable scaling factor regulating nonlinearity.

This activation enables dynamic gating, enabling flexible decision boundaries and improving performance in highly imbalanced intrusion-detection tasks.

Intrusion detection datasets are typically highly imbalanced and exhibit heterogeneous traffic patterns, ranging from near-linear background flows to bursty, heavy-tailed attack behaviors. To address this challenge, the learnable gating mechanism of ACON enables the channel-wise adaptation of nonlinearity, improving gradient flow in sparse regions and capturing subtle decision boundaries associated with rare attack classes.

Comparison with Traditional ReLU.

The fixed ReLU function, $ReLU(x) = max(0, x)$, discards all negative activations and can lead to "dying-neuron" issues in sparse minority regions. In contrast, the ACON activation introduces learnable parameters (p_1, p_2, β) that enables a smooth transition between linear and nonlinear responses. This adaptability preserves weak or negative signals that may represent subtle intrusion patterns. Empirically, replacing ReLU with ACON stabilizes gradient flow and enhances feature expressiveness, particularly for ultra-minority classes such as U2R and Worms. Therefore, ACON serves as a data-adaptive alternative that enhances both convergence and fidelity in the proposed MIDS-GAN framework.

GAN Training Dynamics.

The adversarial training process alternates between two steps:

- Discriminator Update: Minimize L_{D} using real data $x \sim p_{\mathrm{data}}$ and synthetic data $G(z, c)$
- Generator Update: Minimize L_G, focusing on fooling D and generating label-consistent samples

3.4.3. Minimax Training Objective

Beyond the conventional min–max adversarial game, the proposed framework embeds this training process within a divergence minimization formulation, integrating adversarial loss with KL-guided alignment. This formulation ensures that the generator is optimized not only to deceive the discriminator but also to explicitly minimize distributional discrepancies under feature constraints.

$$min_G max_D E_{x \sim p_data}[logD(x)] + E_{z \sim p_z}[log(1 - D(G(z)))] \tag{11}$$

Optimization was performed using the Adam optimizer, a stochastic gradient–based method with tuned learning rates.

To further align with the theoretical foundation of conditional GANs, the explicit formulation of loss functions and KL-divergence used in during training is presented. The discriminator loss is defined as

$$L_D = E_{(x,c) \sim p_{\mathrm{dt}}}[\log D_{\mathrm{source}}(x)] + E_{(z,c) \sim p(z,c)}[\log(1 - D_{\mathrm{source}}(G(z, c)))] - \lambda_{\mathrm{cls}} \cdot L_{\mathrm{class}} \tag{12}$$

The generator loss becomes

$$L_G = E_{(z,c) \sim p(z,c)}[\log D_{\mathrm{source}}(G(z, c))] + \lambda_{\mathrm{cls}} E_{(z,c)}[\log P(c \mid G(z, c))] \tag{13}$$

Distributional similarity between real and generated data is evaluated using KL-divergence (Equation (7)). Training is optimized via gradient backpropagation, with convergence monitored through class-conditional KL-divergence.

Explanation of Algorithm 1.

Algorithm 1 outlines the adversarial training process of the proposed MIDS-GAN, which integrates conditional generation [25], auxiliary classification [26], and the adaptive ACON activation [8]. The process begins with the initialization of generator and discrim-

inator parameters. For each epoch, real samples from the dataset and synthetic samples generated by $G(z, c)$ are alternately used to update the discriminator D, following the mini–max adversarial framework of Goodfellow et al. [15]. The generator is subsequently updated to minimize the adversarial loss while maintaining class-label consistency, consistent with the ACGAN principle [26]. The ACON activation [8] dynamically adapts neuron responses to stabilize gradient flow. Training iterations continue until the class-conditional KL divergence between real and synthetic data distributions converges, ensuring that the generated data are statistically aligned with the minority-class manifold.

Algorithm 1. Training mechanism of ACON-Activated ACGAN with SFS based on minibatch stochastic gradient descent.

- **Input:** $k = 1$ (number of D steps), $\alpha = 2 \times 10^{-4}$ (Adam), $n\cdot = 128$ (minibatch size),

θ_D, θ_G (initialized), SFS-selected features.

- **Output:** Trained generator $G\cdot$ (class-conditioned synthesis).

for number of training iterations **do**
for k steps **do**

- Sample minibatch $\{z_i\}_{i=1}^{m} \sim p_z(z)$
- Sample minibatch $\{(x_i, c_i)\}_{i=1}^{m} \sim p_{\text{data}}(x, c)$
- Compute source loss

$$\mathcal{L}_{\text{source}} = \frac{1}{m}\sum_{i=1}^{m}[\log D_{\text{source}}(x_i) + \log(1 - D_{\text{source}}(G(z_i, c_i)))]$$

- Compute class loss

$$\mathcal{L}_{\text{class}} = \frac{1}{2m}\sum_{i=1}^{m}[\log P(c_i \mid x_i) + \log P(c_i \mid G(z_i, c_i))]$$

- Update D by minimizing

$$\mathcal{L}_{\mathcal{D}} = -\mathcal{L}_{\text{source}} - \lambda_{\text{cls}} \cdot \mathcal{L}_{\text{class}}$$

end for

- Resample $\{(z_i, c_i)\}_{i=1}^{m}$
- Update $G\cdot$ by minimizing

$$L_G = -\frac{1}{m}\sum_{i=1}^{m}\log D_{\text{source}}(G(z_i, c_i)) - \lambda_{\text{cls}}\frac{1}{m}\sum_{i=1}^{m}\log P(c_i \mid G(z_i, c_i))$$

end for

Note: Stop when the KL-divergence between real and synthetic feature distributions converges.

3.5. Rationale for Determining the Number of Synthetic Samples per Minority Class

Severe class imbalance is known to impair generalization in intrusion detection systems [5,42], while oversampling methods such as SMOTE and its variants often overlook redundancy and overfitting risks in pursuit of full balance [9,12]. Recent GAN-based

IDS approaches (e.g., DGM, SPE-ACGAN) have recognized the challenge posed by ultra-minority classes [19,28]. Motivated by these findings, this propose a principled heuristic strategy for determining the number of synthetic samples per minority class, ensuring sufficient representation while avoiding excessive upsampling in rare categories.

In imbalanced intrusion detection datasets such as NSL-KDD and UNSW-NB15, minority classes are typically underrepresented, leading to poor generalization by classifiers. To address this, a principled and heuristic-guided method is adopted to determine the number of synthetic samples (T_c) generated for each minority class (C). The objective is to maintain class representation diversity while avoiding redundancy and overfitting, particularly in extremely rare classes.

The procedure is structured into three main steps, described in the following Sections 3.5.1–3.5.3:

3.5.1. Computation of Minority-to-Majority Ratio

Initially, the raw ratio R_c is computed, which quantifies the relative proportion of a minority class C with respect to the majority class [13]:

$$R_c = \frac{N_c}{N_{\text{major}}} \tag{14}$$

where

N_c is the number of original samples in class C.

N_{major} is the number of samples in the majority class (typically "Normal").

This ratio serves as a baseline metric to understand the degree of imbalance across classes.

3.5.2. Theoretical Target for Full Balance

Assuming perfect balance, each class should contain the same number of samples as the majority. The theoretical multiplier M_c required to achieve full balance is computed as [42]:

$$M_c = \frac{1}{R_c} = \frac{N_{\text{major}}}{N_c} \tag{15}$$

The corresponding target number of synthetic samples is

$$T_c^{\text{ideal}} = M_c \times N_c = N_{\text{major}} \tag{16}$$

However, fully replicating the majority count is often impractical for extremely small classes (e.g., U2R in NSL-KDD, Worms in UNSW-NB15), as it can lead to highly redundant risks generating synthetic data with limited diversity [12,14], thereby increasing the risk of overfitting and degrading downstream classifiers performance.

3.5.3. Adjusted Target with Heuristic Constraints

To determine a more practical target T_c, empirical upper bounds $N_{\text{cap}}^{(c)}$ are introduced for each class based on domain knowledge. The final number of synthetic samples is then defined as

$$T_c = \min\left(M_c N_c,\ N_{\text{cap}}^{(c)}\right) \tag{17}$$

where

$N_{\text{cap}}^{(c)}$ is the empirically chosen cap per class.

This approach balances representational adequacy with the need to avoid excessive upsampling [12,14]. The selection of $N_{\text{cap}}^{(c)}$ is guided by the following criteria:

- **Class complexity:** complex classes, such as Shellcode, require more diversity.

- **Overfitting risk:** extremely rare classes (e.g., U2R or Worms) are capped to prevent synthetic redundancy [19,28].
- **Minimum threshold for classifier training:** each class is ensured to have enough samples (typically >1000) to enable effective learning (Tables 1 and 2).

Table 1. Heuristic-based strategy for minority class synthetic data generation on NSL-KDD.

Class	Training Set N_c	Test Set	Raw Ratio $\frac{N_c}{N_{\text{Normal}}}$	Required Multiplier M_c	Applied Multiplier ($x\ M_c$)	Final Target T_c	Class Type
Normal	67,342	9710	1.0000	1.0	-	0	Major
DoS	45,927	7457	0.6819	1.5	-	0	Major
Probe	11,656	2421	0.1731	5.8	0.43	5000	Minor
R2L	995	2754	0.0148	67.7	3	3000	Minor
U2R	52	200	0.0008	1294.7	38	2000	Rare

Table 2. Heuristic-based strategy for minority class synthetic data generation on UNSW-NB15.

Class	Training Set N_c	Test Set	Raw Ratio $\frac{N_c}{N_{\text{Normal}}}$	Required Multiplier M_c	Applied Multiplier ($x\ M_c$)	Final Target T_c	Class Type
Analysis	2000	667	0.0357	28.0	1.5	3000	Minor
Backdoor	1746	583	0.0312	32.1	1.7	3000	Minor
DoS	12,264	4089	0.2197	4.5	0.24	3000	Minor
Exploits	33,393	11,132	0.5963	1.7	-	0	Major
Fuzzers	18,184	6062	0.3240	3.1	0.27	5000	Minor
Generic	40,000	18,871	0.7143	1.4	-	0	Major
Normal	56,000	37,000	1.0000	1.0	-	0	Major
Reconnaissance	10,491	3494	0.1873	5.3	0.48	5000	Minor
Shellcode	1133	378	0.0202	49.4	2.6	3000	Minor
Worms	130	44	0.0023	430.8	15.4	2000	Rare Avoid overfitting ($\times$15 is sufficient)

3.6. Machine Learning Classifier

To evaluate the effectiveness of the synthetic data generated by the proposed framework, multiple machine learning classifiers were employed as downstream models for intrusion detection. This evaluation aims to determine whether the generated samples enhance the detection capability for minority attack classes, which are typically underrepresented in real-world network traffic datasets.

The evaluation procedure follows a consistent experimental protocol in which all classifiers were trained on both the original dataset and an augmented one combining real and GAN-generated samples. To ensure fair and unbiased comparisons, preprocessing steps include normalization, structured feature selection, and class-balanced sampling.

Model performance is evaluated using weighted F1-score, precision, and recall under k-fold cross-validation (k = 5) to enhance robustness and reduce variance from sampling bias. This experimental design enables a systematic assessment of whether the proposed generative model can improve minority-class detection across different classification algorithms.

Weighted F1-score was chosen as the primary evaluation metric because it accounts for class imbalance by weighing each class according to its sample proportion, thereby providing a fair representation of performance across both majority and minority attack categories. The 5-fold cross-validation (K = 5) setup ensures robust and computationally efficient evaluation, consistent with recent IDS studies [43,44]. Preliminary tests with higher K values produced similar results, confirming the stability and reliability of this configuration.

3.7. Computational Complexity (Concise)

Let N denote the number of samples, B, the batch size; d, the feature dimension; C, the number of classes; and $\mathcal{C}_G, \mathcal{C}_D$ the FLOP cost of one forward pass of the generator and discriminator. With k_D discriminator updates per generator step, one epoch scales as

$$\mathcal{O}\left(\frac{N}{B}\left((2k_D+1)\,C_D + 2\,C_G\right)\right) \tag{18}$$

Dominated by MPL matrix multiplications:

$$\mathcal{O}\left(B\sum_{l=1}^{L} n_{l-1}n_l\right) \tag{19}$$

The proposed additions preserve the asymptotic order: KL-guided alignment introduces a linear-time term $\mathcal{O}(B\,d) - \mathcal{O}(B\,d\,C)$; ACON introduces a minor constant-factor overhead compared with ReLU; and feature constraints are element-wise $\mathcal{O}(B\,d)$. Generating M synthetic samples incur a cost of $\mathcal{O}(M\,C_G)$, corresponding to a forward of G only. During inference, the GAN is not used—only the downstream classifier is executed—thus keeping deployment cost unchanged.

4. Experimental and Results

This section presents the experimental setup, implementation environment, and obtained results of the proposed MIDS-GAN framework. It outlines the datasets, system configuration, and baseline methods, followed by performance evaluation using multiple classifiers. Results are analyzed using accuracy, weighted F1-score, precision, recall, KL divergence, and visualization techniques to demonstrate the effectiveness of MIDS-GAN in mitigating class imbalance and overlap in intrusion detection.

4.1. System and Tool Requirements

To ensure reproducibility and transparency, the implementation environment used for developing and evaluating the proposed MIDS-GAN framework is summarized. All experiments were conducted in Python 3.8 using TensorFlow-GPU as the deep learning backend and on a high-performance workstation. The software stack includes libraries for deep learning, machine learning, and data preprocessing, as listed in Table 3.

Table 3. Implementation Environment.

Component	Specification
Operating System	Ubuntu 20.04 LTS
Development IDE	Visual Studio Code 1.105.1
Programming Lang.	Python 3.x
Deep Learning	TensorFlow-GPU 2.4, Keras 2.4.3
Machine Learning	scikit-learn 0.24.0
Data Processing	Pandas 1.1.5, NumPy 1.19.5
Utilities	tqdm 4.50.0, Matplotlib 3.3, Seaborn 0.11
Hardware	Intel Core i5 CPU, 16 GB RAM, NVIDIA RTX 3060 Ti GPU (8 GB VRAM)

4.2. Dataset Information

This study employs two widely used benchmark datasets: NSL-KDD and UNSW-NB15. Both are publicly available and extensively adopted in intrusion detection research, offering complementary perspectives—NSL-KDD as a refined classical benchmark and UNSW-NB15 as a modern traffic dataset. Detailed descriptions and references are provided in the following subsections.

4.2.1. NSL-KDD

The NSL-KDD dataset is an enhanced and cleaned version of the original KDD'99 benchmark [45]. Its refinement—removing redundant records and mitigating severe class imbalance—was systematically analyzed and documented by Tavallaee et al. [46]. The dataset comprises 41 features representing network-traffic connections, grouped into five major classes: Normal, Denial-of-Service (DoS), Probe, Remote-to-Local (R2L), and User-to-Root (U2R), and is publicly available through the Canadian Institute for Cybersecurity (CIC) [47].

A key limitation of NSL-KDD is its strong class imbalance. While the Normal and DoS dominate, the minority classes (R2L and U2R) contain only a few hundred instances, posing significant challenges for conventional classifiers. Table 4 summarizes the training and testing distributions for each class.

Table 4. Original Class Distribution of NSL-KDD.

Class	Type	Training Set	Test Set	Total	Ratio (%)
Normal	Major	67,342	9710	77,052	51.88%
DoS	Major	45,927	7457	53,384	35.95%
Probe	Minor	11,656	2421	14,077	9.48%
R2L	Minor	995	2754	3749	2.52%
U2R	Rare	52	200	252	0.17%

4.2.2. UNSW-NB15

The UNSW-NB15 dataset was generated in a controlled environment at the Cyber Range Lab of UNSW Canberra using the IXIA PerfectStorm tool [48]. It consists of modern network traffic with 49 features, including both flow-based and content-based attributes, and is considered more representative of real-world attacks compared to NSL-KDD. The dataset is divided into ten classes: Normal, Generic, Exploits, Fuzzers, DoS, Reconnaissance, Analysis, Backdoor, Shellcode, and Worms.

Similarly to NSL-KDD, the UNSW-NB15 dataset exhibits highly skewed class distribution. Dominant classes such as Generic and Exploits make up most of the samples, while minority classes like Worms and Shellcode are extremely underrepresented. These rare classes are crucial for intrusion detection systems but are often overlooked by classifiers due to their scarcity. Table 5 presents detailed statistics for the UNSW-NB15 dataset, which is publicly available from the UNSW Canberra research portal [49].

Table 5. Original Class Distribution of UNSW-NB15.

Class	Type	Training Set	Test Set	Total	Ratio (%)
Analysis	Minor	2000	667	2667	1.04%
Backdoor	Minor	1746	583	2329	0.90%
DoS	Minor	12,264	4089	16,353	6.35%
Exploits	Major	33,393	11,132	44,525	17.28%
Fuzzers	Minor	18,184	6062	24,246	9.41%
Generic	Major	40,000	18,871	58,871	22.85%
Normal	Major	56,000	37,000	93,000	36.09%
Reconnaissance	Minor	10,491	3494	13,985	5.43%
Shellcode	Minor	1133	378	1511	0.59%
Worms	Rare	130	44	174	0.07%

4.2.3. Dataset Challenges (Imbalance and Overlap)

Both NSL-KDD and UNSW-NB15 pose two critical challenges that hinder intrusion detection performance: class imbalance and class overlap. In NSL-KDD, most records belong to Normal and DoS categories, while minority classes such as R2L and U2R account for less

than 5% of the samples, causing classifiers to favor majority classes and overlook rare but critical intrusions. A similar imbalance exists in UNSW-NB15, where Generic and Exploits dominate, and rare classes like Worms and Shellcode have only a few hundred instances.

Class overlap further complicates UNSW-NB15, as attack categories such as Exploits, Generic, and Fuzzers share similar traffic features, leading to ambiguous decision boundaries and frequent misclassification. Consequently, models trained on such data often fail to generalize effectively to unseen attack variants.

In short, these challenges highlight the need for advanced data augmentation strategies that can both balance class representation and enhance feature separability across intrusion categories.

Dataset scope. NSL-KDD and UNSW-NB15 are widely used IDS benchmarks, representing conventional (primarily IPv4) enterprise and LAN traffic. They do not explicitly capture SDN control/data-plane behaviors or IPv6 flows; therefore, the empirical findings of this study are interpreted within this scope.

4.2.4. Relevance to This Study

The coexistence of class imbalance and overlap makes NSL-KDD and UNSW-NB15 rigorous and suitable benchmarks for evaluating the proposed MIDS-GAN framework. Unlike traditional oversampling methods that simply replicate minority samples, MIDS-GAN is designed to (i) balance minority class representation, (ii) preserve distributional similarity to real data, and (iii) enhance classifier performance in detecting rare intrusions while mitigating misclassification caused by overlapping features.

By simultaneously addressing imbalance and overlap, the proposed framework targets two critical challenges in intrusion detection research. The selected datasets thus serve as realistic testbeds for evaluating the effectiveness of MIDS-GAN in enhancing minority intrusion detection within complex and heterogeneous network environments.

4.3. Implementation Details

To ensure reproducibility, this section summarizes the implementation settings used for training and evaluating the proposed MIDS-GAN. It details the GAN architecture, training hyperparameters, and evaluation protocols, including latent dimension, batch size, optimizer configuration, and the number of training epochs. Additionally, the settings of baseline oversampling methods and classifier models are kept consistent to ensure a fair comparison.

4.4. Model Hyperparameter Configuration

The architectural and training configurations of MIDS-GAN were carefully selected to balance stability, convergence, and expressiveness in modeling minority class distributions. Table 6 summarizes the key hyperparameters used across four experimental setups—ACON and ReLU activations on both NSL-KDD and UNSW-NB15 datasets. These configurations were consistently applied in all GAN training phases to ensure comparability and reproducibility.

Justification of Hyperparameters

The hyperparameter selection for MIDS-GAN was guided by empirical best practices and the convergence behavior of Kullback–Leibler (KL) divergence between real and synthetic data distributions. Rather than manual tuning of downstream classifiers, the focus was placed on GAN-side configurations that directly affect the fidelity of minority class generation. The complete configuration is presented in Table 6.

Table 6. Hyperparameters of MIDS-GAN variants.

Parameter	NSL–ACON	NSL–ReLU	UNSW–ACON	UNSW–ReLU
Epochs	2000	2000	2000	2000
Batch size	128	128	128	128
Latent dim (z)	100	100	100	100
Optimizer (G/D)	Adam	Adam	Adam	Adam
LR (G/D)	2×10^{-4}	2×10^{-4}	2×10^{-4}	2×10^{-4}
Layers (G/D)	4/4	4/4	4/4	4/4
Hidden units (G)	$128 \rightarrow 64 \rightarrow 32 \rightarrow (*d_{NSL})$	$128 \rightarrow 64 \rightarrow 32 \rightarrow (d_{NSL})$	$128 \rightarrow 64 \rightarrow 32 \rightarrow (*d_{UNSW})$	$128 \rightarrow 64 \rightarrow 32 \rightarrow (d_{UNSW})$
Hidden units (D)	$128 \rightarrow 64 \rightarrow 32 \rightarrow (1 + C_{NSL})$	$128 \rightarrow 64 \rightarrow 32 \rightarrow (1 + C_{NSL})$	$128 \rightarrow 64 \rightarrow 32 \rightarrow (1 + C_{UNSW})$	$128 \rightarrow 64 \rightarrow 32 \rightarrow (1 + C_{UNSW})$
Activation	ACON (trainable α,β,γ)	ReLU	ACON (trainable α,β,γ)	ReLU
Dropout	0.2	0.2	0.2	0.2
Update Ratio (D:G)	1:3	1:3	1:3	1:3
Validation	5-fold CV	5-fold CV	5-fold CV	5-fold CV

* Note: d_{NSL} and d_{UNSW} denote the feature dimensions of the respective datasets. The hyperparameters listed in Table 6 were selected based on commonly adopted configurations in GAN-related studies [15,19,24] and were empirically refined for convergence stability on NSL-KDD and UNSW-NB15 datasets.

All models used a latent dimension of 100, batch size of 128, and a discriminator-to-generator update ratio of 1:3. The latent dimension was set to 100, following common practice in GAN research, to balance expressive capacity and training stability. A batch size of 128 was selected to optimize computational efficiency and stable gradient estimation. Training was conducted for 2000 epochs, empirically determined from the KL divergence curves, which steadily decreased and plateaued around epochs 1800–2000 (Figures 3 and 4). This indicates sufficient generator convergence for learning of minority-class patterns in highly imbalanced datasets without further oscillation or collapse.

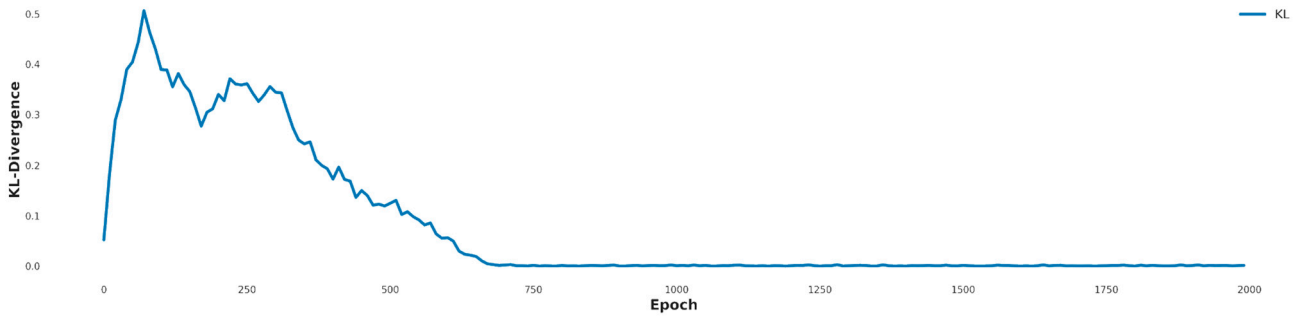

Figure 3. KL divergence across training epochs for NSL-KDD dataset. The ACON-based MIDS-GAN shows faster convergence and lower divergence than the ReLU-based model, particularly benefiting rare classes such as R2L and U2R.

The generator and discriminator were optimized using Adam with learning rates of 0.0002 and 0.0001, respectively, to stabilize adversarial training and prevent mode collapse, given the discriminator's faster convergence. The proposed model employed ACON (with trainable parameters α, β, γ) to enhance non-linearity and adaptability, while ReLU was included as a baseline for comparison. This adaptive activation approach is conceptually aligned with strategies such as SPOCU, addressing limitations of fixed activations like ReLU. KL divergence plots confirmed that ACON-based models converged faster and more smoothly, especially for challenging classes such as R2L and U2R (NSL-KDD) and Worms and Shellcode (UNSW-NB15). This KL-guided convergence thus functions both as a stopping criterion and as an indirect indicator of synthetic data quality and diversity.

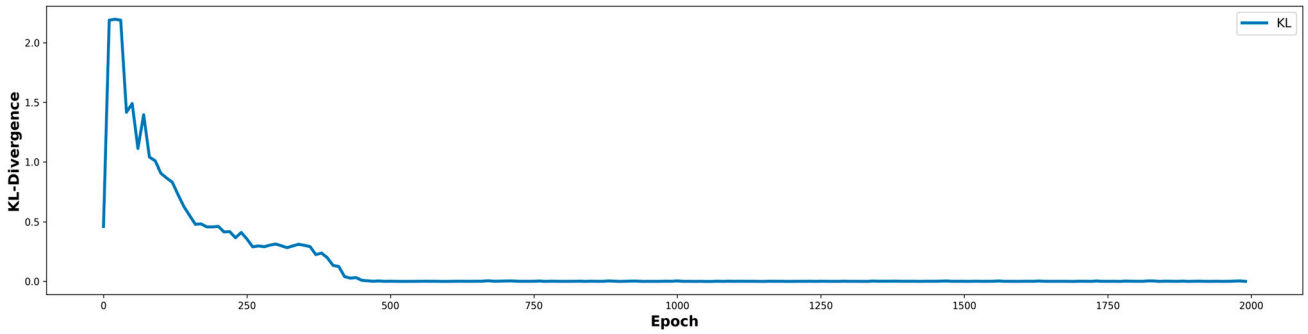

Figure 4. KL divergence across training epochs for the UNSW-NB15 dataset. ACON-based models achieve lower divergence and better alignment of minority class, particularly in challenging categories like Shellcode and Worms.

Evaluation metrics included weighted F1-score, precision, recall, and KL divergence to provide a comprehensive assessment of performance, particularly under class imbalance. A 5-fold cross-validation was used to ensure the reliability and robustness of the results.

Overall, these hyperparameter choices, supported by divergence-based monitoring, enabled MIDS-GAN to generate class-balanced synthetic samples that closely reflect the real data distribution, thus enhancing both data utility and downstream classifier performance.

4.5. Evaluation Criteria

To evaluate the effectiveness of the generated synthetic data and the classifiers used in this study, four widely recognized performance metrics were used: precision, recall, F1-score, and weighted F1-score. Derived from the confusion matrix, these metrics are essential for assessing detection capability, particularly under class imbalance.

The precision metric measures the proportion of correctly predicted positive instances relative to all predicted positives:

$$Precision = \frac{TP}{TP + FP} \tag{20}$$

The recall metric, also known as sensitivity, quantifies the proportion of actual positives correctly identified:

$$Recall = \frac{TP}{TP + FN} \tag{21}$$

The **F1-score** is the harmonic mean of precision and recall, balancing their contributions when evaluating models under class imbalance:

$$F1 - score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{22}$$

To address class imbalance across multiple labels, the weighted F1-score was employed. This metric accounts for the support (sample count) of each class i when averaging:

$$\text{Weighted } F1 = \frac{\sum_{i=1}^{K} \text{Support}_i \cdot F1_i}{\text{Total}} \tag{23}$$

where

- TP = True Positives.
- FP = False Positives.
- FN = False Negatives.
- Support_i = Number of true instances in class i

- $F1_i$ = F1-score of class i
- K = Total number of classes.

To ensure the generalizability and statistical robustness of the evaluation, 5-fold cross-validation ($K = 5$) was employed. This approach minimizes bias from any specific train-test split and verifies that performance improvements from synthetic data generation remain consistent across folds.

4.6. Comparative Results and Effectiveness of MIDS-GAN

To validate the effectiveness of the proposed MIDS-GAN, its performance was compared with widely used oversampling baselines (SMOTE, ADASYN, SMOTEENN, and Borderline-SMOTE) across two benchmark datasets: NSL-KDD and UNSW-NB15. Evaluation was conducted using multiple classifiers (Random Forest, Decision Tree, Neural Network, and SVM) based on precision, recall, F1-score, and weighted F1. Particular attention was placed on minority classes (R2L, U2R, Shellcode, Worms), which represent the most challenging scenarios in intrusion detection.

4.6.1. NSL-KDD Results

As shown in Table 7, MIDS-GAN achieved notable improvement in detecting challenging classes. For example, R2L recall increased from 2% to 27% with Random Forest, and U2R recall improved from 1% to 17% with SVM. These results demonstrate that MIDS-GAN effectively captures the underlying distribution of rare classes, outperforming oversampling methods. Meanwhile, majority classes such as DoS and Normal maintained strong performance, indicating that enhanced minority detection did not compromise majority-class accuracy.

Table 7. Classification performance of classifiers on NSL-KDD with and without MIDS-GAN.

		Precision (%)	**Recall (%)**	**F1 Score (%)**	**Precision (%)**	**Recall (%)**	**F1 Score (%)**
Class	**Classifier**	**Original Data**			**MIDS**		
DoS	Random forest	96	78	86	93	87	90
DoS	Decision tree	95	79	86	90	75	82
DoS	Neural network	96	81	88	92	83	87
DoS	SVM	96	78	86	95	82	88
Probe	Random forest	84	62	72	44	86	58
Probe	Decision tree	81	65	72	41	86	56
Probe	Neural network	73	60	66	43	83	56
Probe	SVM	80	2	67	42	87	57
R2L	Random forest	96	2	5	67	27	39
R2L	Decision tree	71	5	9	41	86	55
R2L	Neural network	85	9	16	43	82	56
R2L	SVM	97	8	16	42	87	57
U2R	Random forest	62	1	3	60	17	26
U2R	Decision tree	60	6	11	44	23	30
U2R	Neural network	72	8	15	61	30	40
U2R	SVM	57	1	2	35	26	30

4.6.2. 2 UNSW-NB15 Results

The UNSW-NB15 dataset poses greater challenges due to pronounced class overlap. Nevertheless, MIDS-GAN achieved significant improvements for minority-class detection. For example, Worms recall increased from 18% to 44% with Random Forest, and Shellcode recall improved from 69% to 75% with Decision Tree (Table 8). In contrast, traditional

oversampling methods often produced marginal or even negative effects on these rare categories. Notably, majority classes such as Generic and Exploits retained high F1-scores (above 95%), indicating that synthetic augmentation did not degrade classifier performance on dominant classes.

Table 8. Classification performance of classifiers on UNSW-NB15 with and without MIDS-GAN.

		Precision (%)	Recall (%)	F1 Score (%)	Precision (%)	Recall (%)	F1 Score (%)
Class	**Classifier**	**Original Data**			**MIDS**		
Analysis	Random forest	0	0	0	0	0	0
Analysis	Decision tree	1	7	3	2	8	4
Analysis	Neural network	0	0	0	0	0	0
Analysis	SVM	0	0	0	0	0	0
Backdoor	Random forest	2	12	3	2	1	4
Backdoor	Decision tree	2	10	3	3	1	5
Backdoor	Neural network	2	1	1	3	2	2
Backdoor	SVM	6	5	11	8	1	2
DoS	Random forest	57	10	17	28	33	30
DoS	Decision tree	29	19	23	27	30	28
DoS	Neural network	28	5	9	31	64	42
DoS	SVM	32	34	33	32	70	44
Exploits	Random forest	60	79	68	78	66	71
Exploits	Decision tree	62	68	65	73	61	66
Exploits	Neural network	56	88	68	74	64	69
Exploits	SVM	81	49	61	83	52	64
Fuzzers	Random forest	30	57	39	86	70	77
Fuzzers	Decision tree	27	47	35	91	65	75
Fuzzers	Neural network	31	52	39	80	61	69
Fuzzers	SVM	22	51	31	71	50	59
Generic	Random forest	100	96	98	100	96	98
Generic	Decision tree	98	97	98	99	97	98
Generic	Neural network	99	95	97	99	95	97
Generic	SVM	99	94	97	99	95	97
Reconnaissance	Random forest	92	80	86	89	82	86
Reconnaissance	Decision tree	89	78	83	88	80	84
Reconnaissance	Neural network	68	82	75	80	81	80
Reconnaissance	SVM	44	79	57	50	85	63
Shellcode	Random forest	39	71	50	40	75	52
Shellcode	Decision tree	36	69	47	38	71	49
Shellcode	Neural network	22	64	33	30	59	40
Shellcode	SVM	7	69	14	16	37	22
Worms	Random forest	57	18	27	69	44	53
Worms	Decision tree	52	52	52	45	63	52
Worms	Neural network	50	15	24	43	10	17
Worms	SVM	4	72	7	7	16	25

4.6.3. Weighted F1-Score Comparison

To provide a dataset-wide perspective, Tables 9 and 10 summarize the weighted F1-scores across all classifiers and augmentation methods.

Table 9. Weighted F1-scores on NSL-KDD.

Model	Random Forest	Decision Tree	Neural Network	SVM
Original	70	71	71	68
SMOTE	71	72	70	71
ADASYN	72	66	70	71
SMOTEENN	73	70	71	71
Borderline-SMOTE	73	70	70	71
MIDS-ACON	72	69	70	71
MIDS-RELU	72	71	70	71

Table 10. Weighted F1-scores on UNSW-NB15.

Model	Random Forest		Decision Tree		Neural Network		SVM	
Original	76		76		67		58	
SMOTE	77		75		73		70	
ADASYN	77		75		73		69	
SMOTEENN	73		73		67		68	
Borderline-SMOTE	77		75		72		69	
MIDS-ACON	78		77		77		73	
MIDS-RELU	71		77		77		73	
Weighted F1-scores	UNSW-NB15	NSL-KDD	UNSW-NB15	NSL-KDD	UNSW-NB15	NSL-KDD	UNSW-NB15	NSL-KDD
Original	76	70	76	71	67	71	58	68
SMOTE	77	71	75	72	73	70	70	71
ADASYN	77	72	75	66	73	70	69	71
SMOTEENN	73	73	73	70	67	71	68	71
Borderline-SMOTE	77	73	75	70	72	70	69	71
MIDS-ACON	**78**	**72**	**77**	**69**	**77**	**70**	**73**	**71**
MIDS-RELU	**71**	**72**	**77**	**71**	**77**	**70**	**73**	**71**

The weighted F1-score comparisons confirm that MIDS-GAN achieves performance comparable to or better than conventional oversampling baselines. On NSL-KDD, results were stable across classifiers, with MIDS-ACON matching or slightly exceeding SMOTE and Borderline-SMOTE. On UNSW-NB15, MIDS-ACON consistently outperformed baseline methods in our evaluations, reaching 78% weighted F1 with Random Forest, exceeding both SMOTE (77%) and ADASYN (77%). These findings indicate that the adaptive activation (ACON) provides a measurable advantage over ReLU by modeling more complex decision boundaries.

Overall, the weighted F1 performance across all classifiers and augmentation methods (Tables 9 and 10) demonstrates dataset-wide improvements, complementing the class-specific recall gains observed for minority attack categories.

Tables 11 and 12 highlight the recall improvements for key minority classes in the NSL-KDD and UNSW-NB15 datasets. The results demonstrate substantial performance gains for rare classes such as R2L, U2R, Shellcode, and Worms, indicating the effectiveness of MIDS-GAN augmentation.

Table 11. ΔRecall on ultra-minority classes (NSL-KDD and UNSW-NB15).

Dataset	Class	Classifier	Recall (Original) (%)	Recall (MIDS-GAN) (%)	ΔRecall (%)
NSL-KDD	R2L	Random Forest	2	27	25
NSL-KDD	U2R	SVM	1	26	25
UNSW-NB15	Worms	Random Forest	18	44	26

Table 12. Cross-study ΔRecall (%) on ultra-minority classes (best classifier per study). ("Advantage" = MIDS-GAN − DGM, in percentage points.)

Dataset	Class	MIDS-GAN	Best Classifier	DGM [19]	Best Classifier	Advantage (pp)
NSL-KDD	R2L	+25 (2→27)	Random Forest	+17 (9→26)	SVM	+8
NSL-KDD	U2R	+25 (1→26)	SVM	+8 (9→17)	Random Forest	+17
UNSW-NB15	Worms	+26 (18→44)	Random Forest	+1 (0→1)	Random Forest	+25
Average Improvement (Δ pp)						≈+16.7

Note. Due to variations in experimental protocols across studies, these figures are presented for contextual comparison rather than a direct ranking.

Minority-class sensitivity. While the weighted F1 summarizes overall performance, Table 11 highlights the largest ΔRecall on rare but critical classes—R2L, U2R, and Worms—demonstrating substantial gains for ultra-minority detection. Additionally, Table 12 presents a cross-study comparison of ΔRecall (%) for ultra-minority classes between MIDS-GAN and the recently proposed Data Generative Model (DGM) [19], highlighting MIDS-GAN's superior effectiveness in detecting ultra-minority categories.

5. Discussion

5.1. Portability and Scope

MIDS-GAN, designed for tabular flow and content features, is inherently adaptable to other network environments (e.g., IPv6, SDN, 5G and beyond) given suitable protocol-specific features and labeled data, with retraining and validation in those contexts. However, current evidence is limited to NSL-KDD and UNSW-NB15, and no out-of-scope performance is claimed.

5.2. Limitations

The evaluated datasets (NSL-KDD, UNSW-NB15) primarily represent conventional IPv4 enterprise and LAN traffic and do not capture model SDN control/data-plane dynamics or IPv6 flows. Consequently, external validity to IPv6, SDN, and 5G-and-beyond environments remains to be established.

5.3. Significance of Recall Improvement on Ultra-Minority Classes

While most GAN-based IDS studies focus on overall F1-score, such aggregate metrics can obscure the detection behavior of ultra-minority classes (e.g., U2R, R2L, Worms, Shellcode). In contrast, Recall directly reflects a model's ability to detect true attacks, where even small shortfalls may result in undetected intrusions. The proposed MIDS-GAN achieved consistent 25–26 percentage points (pp) Recall improvement across both datasets, outperforming the DGM framework [19] by an average margin of ≈16.7 pp (Tables 11 and 12). The results demonstrate the framework's effectiveness in expanding minority-class manifolds and reducing false negatives.

Additionally, the enhanced Recall underscores MIDS-GAN's security-oriented robustness, driven by its integration of ACON-based adaptive activations and KL-divergence

alignment, which collectively improve class separability, and reduce false negatives under severe class imbalance. Baseline data distributions prior to augmentation (Appendix A) further confirm that MIDS-GAN effectively expands sparse and overlapping minority classes, thereby improving both detection reliability and generalization.

6. Conclusions

This study introduced MIDS-GAN, a divergence-guided augmentation framework that enhances distributional alignment between real and synthetic network traffic for intrusion detection. Evaluation on NSL-KDD and UNSW-NB15 demonstrated that MIDS-GAN significantly improved the detection of ultra-minority classes (e.g., R2L/U2R; Shellcode/Worms) without compromising majority-class performance. The findings emphasize that class-conditional generation with KL-guided training and ACON activations provides a principled and effective alternative to heuristic oversampling such as SMOTE, ADASYN, SMOTEENN, and Borderline-SMOTE.

Future work will extend framework's generalizability by evaluating MIDS-GAN on SDN, IPv6, and 5G datasets incorporating protocol-appropriate features and broader telemetry. Further investigation will assess robustness under domain and temporal shifts, conduct targeted ablations, and evaluate computational efficiency for practical deployment.

Author Contributions: Conceptualization, C.K., P.B. and P.W.; Formal Analysis, C.K. and P.B.; Methodology, C.K. and P.W.; Resources, C.K., P.B. and P.W.; Validation, C.K., P.B. and P.W.; Supervision, P.B. and P.W.; Project administration, P.W.; Software, C.K.; Writing—original draft, C.K.; Writing—review and editing, P.B. and P.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets used in this study (NSL-KDD and UNSW-NB15) are publicly available. NSL-KDD can be accessed at http://www.unb.ca/cic/datasets/nsl.html (accessed on 12 September 2025), and UNSW-NB15 at https://research.unsw.edu.au/projects/unsw-nb15-dataset (accessed on 12 September 2025).

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. t-SNE Visualization of Real Data

To provide a visual reference of the intrinsic class structure in the benchmark datasets, we present two-dimensional t-SNE projections of the original NSL-KDD and UNSW-NB15 data (Figures A1 and A2). Prior to t-SNE, all features were normalized and optionally reduced to 50 principal components (PCA) for noise suppression.

These plots illustrate the natural class distributions and inter-class overlap present in the unaugmented datasets, serving as a baseline for comparison with the quantitative results reported in Sections 4.4–4.6.

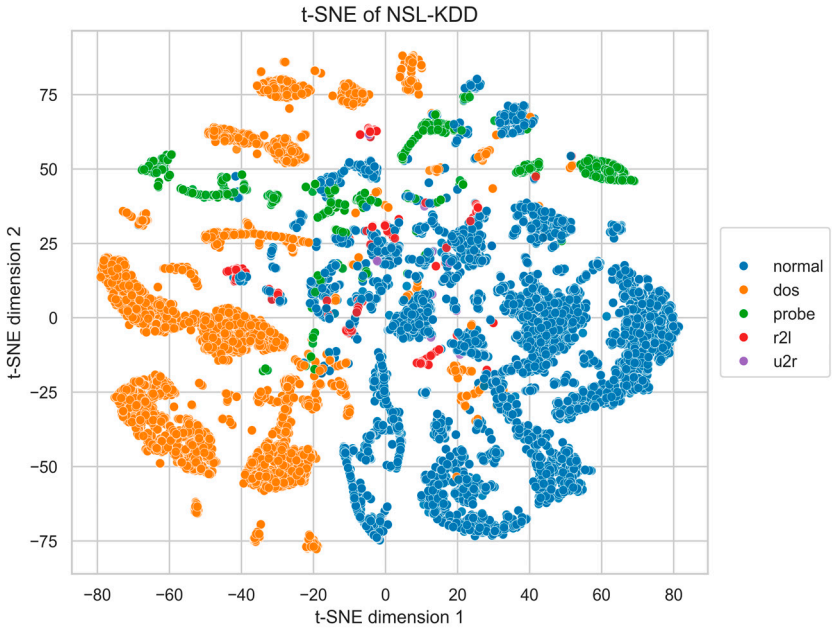

Figure A1. t-SNE projection of NSL-KDD real data and MIDS-GAN-generated.

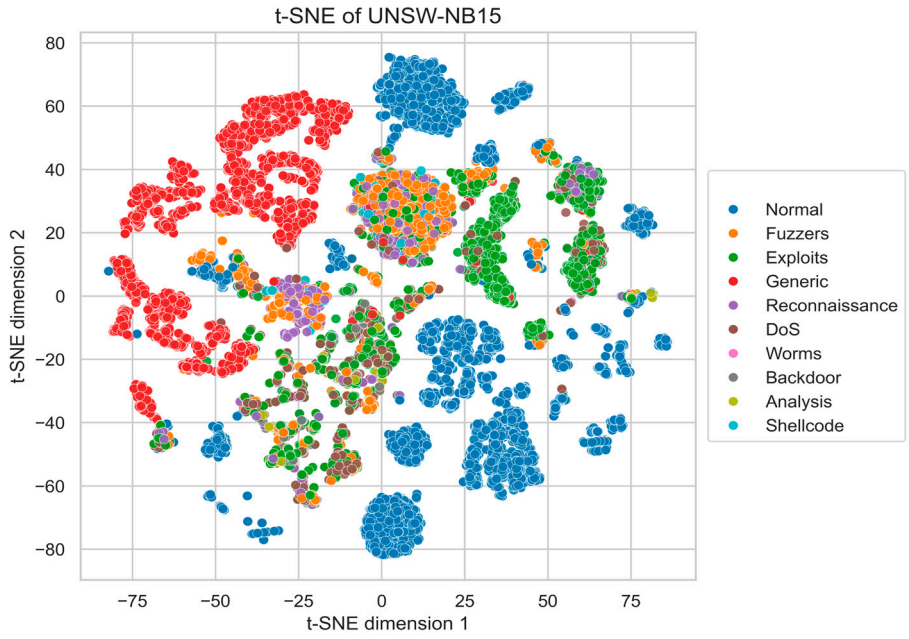

Figure A2. t-SNE projection of UNSW-NB15 real data and MIDS-GAN-generated.

References

1. Buczak, A.L.; Guven, E. A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 1153–1176. [CrossRef]
2. Ring, M.; Wunderlich, S.; Scheuring, D.; Landes, D.; Hotho, A. A Survey of Network-Based Intrusion Detection Data Sets. *Comput. Secur.* **2019**, *86*, 147–167. [CrossRef]
3. 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*, 76071–76093. [CrossRef]
4. Thakkar, A.; Lohiya, R. A Review on Challenges and Future Research Directions for Machine Learning-Based Intrusion Detection System. *Arch. Comput. Methods Eng.* **2023**, *30*, 4245–4269. [CrossRef]
5. Goldschmidt, P.; Chudá, D. Network Intrusion Datasets: A Survey, Limitations, and Recommendations. *Comput. Secur.* **2025**, *156*, 104510. [CrossRef]
6. Park, C.; Lee, J.; Kim, Y.; Park, J.-G.; Kim, H.; Hong, D. An Enhanced AI-Based Network Intrusion Detection System Using Generative Adversarial Networks. *IEEE Internet Things J.* **2023**, *10*, 2330–2345. [CrossRef]

7. Apicella, A.; Donnarumma, F.; Isgrò, F.; Prevete, R. A Survey on Modern Trainable Activation Functions. *Neural Netw.* **2021**, *138*, 14–32. [CrossRef]
8. Ma, N.; Zhang, X.; Liu, M.; Sun, J. Activate or Not: Learning Customized Activation. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 8028–8038. [CrossRef]
9. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic Minority Over-Sampling Technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. [CrossRef]
10. He, H.; Bai, Y.; Garcia, E.A.; Li, S. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In Proceedings of the 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), Hong Kong, China, 1–8 June 2008; pp. 1322–1328. [CrossRef]
11. Batista, G.; Prati, R.; Monard, M.-C. A study of the behavior of several methods for balancing machine learning training data. *SIGKDD Explor. Newsl.* **2004**, *1*, 20–29. [CrossRef]
12. Han, H.; Wang, W.Y.; Mao, B.H. Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning. In *Advances in Intelligent Computing*; Huang, D.S., Zhang, X.P., Huang, G.B., Eds.; ICIC 2005. Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2005; Volume 3644. [CrossRef]
13. Napierala, K.; Stefanowski, J. Types of minority class examples and their influence on learning classifiers from imbalanced data. *J. Intell. Inf. Syst.* **2016**, *46*, 563–597. [CrossRef]
14. Guo, J.; Wu, H.; Chen, X.; Lin, W. Adaptive SV-Borderline SMOTE-SVM Algorithm for Imbalanced Data Classification. *Appl. Soft Comput.* **2024**, *150*, 110986. [CrossRef]
15. Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. In Proceedings of the 28th International Conference on Neural Information Processing Systems (NIPS'14), Montreal, QC, Canada, 8–13 December 2014; MIT Press: Cambridge, MA, USA, 2014; Volume 2, pp. 2672–2680. Available online: https://proceedings.neurips.cc/paper/2014/file/f033ed80deb0234979a61f95710dbe25-Paper.pdf (accessed on 10 September 2025).
16. Yin, C.; Zhu, Y.; Fei, J.; He, X. A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks. *IEEE Access* **2017**, *5*, 21954–21961. [CrossRef]
17. Yang, J.; Li, T.; Liang, G.; He, W.; Zhao, Y. A Simple Recurrent Unit Model Based Intrusion Detection System With DCGAN. *IEEE Access* **2019**, *7*, 83286–83296. [CrossRef]
18. Shahriar, M.H.; Haque, N.I.; Rahman, M.A.; Alonso, M. G-IDS: Generative Adversarial Networks Assisted Intrusion Detection System. In Proceedings of the 2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC), Madrid, Spain, 13–17 July 2020; pp. 376–385. [CrossRef]
19. Dlamini, G.; Fahim, M. DGM: A data generative model to improve minority class presence in anomaly detection domain. *Neural Comput. Appl.* **2021**, *33*, 13635–13646. [CrossRef]
20. Alotaibi, F.; Rassam, M. Adversarial Machine Learning Attacks against Intrusion Detection Systems: A Survey on Strategies and Defense. *Future Internet* **2023**, *15*, 62. [CrossRef]
21. Klinkhamhom, C.; Wuttidittachotti, P.; Boonyopakorn, P. Comparative Evaluation of GAN, CGAN, and ACGAN for Intrusion Detection in Cybersecurity. In Proceedings of the 2025 International Technical Conference on Circuits/Systems, Computers, and Communications (ITC-CSCC), Seoul, Republic of Korea, 7–10 July 2025; pp. 1–4. [CrossRef]
22. Lee, G.-C.; Li, J.-H.; Li, Z.-Y. A Wasserstein Generative Adversarial Network–Gradient Penalty-Based Model with Imbalanced Data Enhancement for Network Intrusion Detection. *Appl. Sci.* **2023**, *13*, 8132. [CrossRef]
23. Gul, S.; Arshad, S.; Saeed, S.M.U.; Akram, A.; Azam, M.A. WGAN-DL-IDS: An Efficient Framework for Intrusion Detection System Using WGAN, Random Forest, and Deep Learning Approaches. *Computers* **2025**, *14*, 4. [CrossRef]
24. Strickland, C.; Zakar, M.; Saha, C.; Soltani Nejad, S.; Tasnim, N.; Lizotte, D.J.; Haque, A. DRL-GAN: A Hybrid Approach for Binary and Multiclass Network Intrusion Detection. *Sensors* **2024**, *24*, 2746. [CrossRef]
25. Mirza, M.; Osindero, S. Conditional Generative Adversarial Nets. *arXiv* **2014**, arXiv:1411.1784. [CrossRef]
26. Odena, A.; Olah, C.; Shlens, J. Conditional Image Synthesis with Auxiliary Classifier GANs. In Proceedings of the 34th International Conference on Machine Learning (ICML 2017), Sydney, Australia, 6–11 August 2017; Precup, D., Teh, Y.W., Eds.; JMLR.org: Sydney, Australia, 2017; pp. 2642–2651.
27. Zhang, K.; Qin, H.; Jin, Y.; Wang, H.; Yu, X. Auxiliary Classifier Generative Adversarial Network Assisted Intrusion Detection System. In Proceedings of the 2022 4th International Conference on Intelligent Information Processing (IIP), Guangzhou, China, 14–16 October 2022; pp. 307–311. [CrossRef]
28. Yang, H.; Xu, J.; Xiao, Y.; Hu, L. SPE-ACGAN: A Resampling Approach for Class Imbalance Problem in Network Intrusion Detection Systems. *Electronics* **2023**, *12*, 3323. [CrossRef]
29. Dahouda, M.K.; Joe, I. A Deep-Learned Embedding Technique for Categorical Features Encoding. *IEEE Access* **2021**, *9*, 114381–114391. [CrossRef]
30. Borisov, V.; Leemann, T.; Seßler, K.; Haug, J.; Pawelczyk, M.; Kasneci, G. Deep Neural Networks and Tabular Data: A Survey. *IEEE Trans. Neural Netw. Learn. Syst.* **2024**, *35*, 7499–7519. [CrossRef]

31. Xu, L.; Skoularidou, M.; Cuesta-Infante, A.; Veeramachaneni, K. Modeling tabular data using conditional GAN. In Proceedings of the 33rd International Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; Curran Associates Inc.: Red Hook, NY, USA, 2019; pp. 7335–7345. [CrossRef]
32. Zhao, Z.; Kunar, A.; Birke, R.; Van der Scheer, H.; Chen, L.Y. CTAB-GAN+: Enhancing tabular data synthesis. *Front. Big Data* **2024**, *6*, 1296508. [CrossRef]
33. Allagi, S.; Pawan, T.; Leong, W.Y. Enhanced Intrusion Detection Using Conditional-Tabular-Generative-Adversarial-Network-Augmented Data and a Convolutional Neural Network: A Robust Approach to Addressing Imbalanced Cybersecurity Datasets. *Mathematics* **2025**, *13*, 1923. [CrossRef]
34. Benesty, J.; Chen, J.; Huang, Y.; Cohen, I. Pearson Correlation Coefficient. In *Noise Reduction in Speech Processing*; Springer Topics in Signal Processing; Springer: Berlin/Heidelberg, Germany, 2009; Volume 2.
35. Khan, J.; Elfakharany, R.; Saleem, H.; Pathan, M.; Shahzad, E.; Dhou, S.; Aloul, F. Can Machine Learning Enhance Intrusion Detection to Safeguard Smart City Networks from Multi-Step Cyberattacks? *Smart Cities* **2025**, *8*, 13. [CrossRef]
36. Sabilla, M.S.; Rahmat, B.; Maulana, D.S. Optimizing Threshold Using Pearson Correlation for Selecting Features of Electronic Nose Signals. *Int. J. Intell. Eng. Syst.* **2019**, *12*, 82–90. [CrossRef]
37. Hall, M.A. Correlation-Based Feature Selection for Machine Learning. Ph.D. Thesis, University of Waikato, Hamilton, New Zealand, 1999. Available online: https://ml.cms.waikato.ac.nz/publications/1999/99MH-Thesis.pdf (accessed on 2 September 2025).
38. Nasir, I.M.; Khan, M.A.; Yasmin, M.; Shah, J.H.; Gabryel, M.; Scherer, R.; Damaševičius, R. Pearson Correlation-Based Feature Selection for Document Classification Using Balanced Training. *Sensors* **2020**, *20*, 6793. [CrossRef]
39. Ozkan-Okay, M.; Samet, R.; Aslan, Ö.; Kosunalp, S.; Iliev, T.; Stoyanov, I. A Novel Feature Selection Approach to Classify Intrusion Attacks in Network Communications. *Appl. Sci.* **2023**, *13*, 11067. [CrossRef]
40. Yin, Y.; Jang-Jaccard, J.; Xu, W.; Singh, A.; Zhu, J.; Sabrina, F.; Kwak, J. IGRF-RFE: A hybrid feature selection method for MLP-based network intrusion detection on UNSW-NB15 dataset. *J. Big Data* **2023**, *10*, 15. [CrossRef]
41. Kullback, S.; Leibler, R.A. On Information and Sufficiency. *Ann. Math. Stat.* **1951**, *22*, 79–86. [CrossRef]
42. He, H.; Garcia, E.A. Learning from Imbalanced Data. *IEEE Trans. Knowl. Data Eng.* **2009**, *21*, 1263–1284. [CrossRef]
43. Rajathi, C.; Rukmani, P. AccFIT-IDS: Accuracy-Based Feature Inclusion Technique for Intrusion Detection System. *Syst. Sci. Control Eng.* **2025**, *13*, 2460429. [CrossRef]
44. Psychogyios, K.; Papadakis, A.; Bourou, S.; Nikolaou, N.; Maniatis, A.; Zahariadis, T. Deep Learning for Intrusion Detection Systems (IDSs) in Time Series Data. *Future Internet* **2024**, *16*, 73. [CrossRef]
45. Dua, D.; Graff, C. KDD Cup 1999 Data Set. UCI Machine Learning Repository. Available online: https://archive.ics.uci.edu/dataset/130/kdd+cup+1999+data (accessed on 4 September 2025).
46. Tavallaee, M.; Bagheri, E.; Lu, W.; Ghorbani, A.A. A detailed analysis of the KDD CUP 99 data set. In Proceedings of the 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, Ottawa, ON, Canada, 8–10 July 2009; pp. 1–6. [CrossRef]
47. Canadian Institute for Cybersecurity (CIC). NSL-KDD Dataset. Available online: https://www.unb.ca/cic/datasets/nsl.html (accessed on 12 September 2025).
48. Moustafa, N.; Slay, J. UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In Proceedings of the 2015 Military Communications and Information Systems Conference (MilCIS), Canberra, Australia, 10–12 November 2015; pp. 1–6. [CrossRef]
49. UNSW Canberra. UNSW-NB15 Dataset. Available online: https://research.unsw.edu.au/projects/unsw-nb15-dataset (accessed on 4 September 2025).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.