A. Lara-Gutierrez, C. Fernandez-Gago and J. A. Onieva, "Adversarial red teaming and imbalance correction in heterogeneous NIDS: A class-specific adaptive GAN framework", *Journal of Information Security and Applications*, vol. 102, pp. 104584, 2026.
http://doi.org/10.1016/j.jisa.2026.104584
NICS Lab. Publications: https://www.nics.uma.es/publications

Adversarial Red Teaming and Imbalance Correction in Heterogeneous NIDS: A Class-Specific Adaptive GAN Framework

Antonio Lara-Gutierrez [a,*], Carmen Fernandez-Gago [a], Jose A. Onieva [a]

[a] *Network, Information and Computer Security (NICS) Lab, University of Málaga, Málaga, 29010, Spain*

Abstract

Modern network infrastructures face a structural long-tail imbalance where malicious events are statistically negligible against benign traffic. This bias causes Machine Learning-based Network Intrusion Detection Systems (NIDS) to systematically overlook rare, high-severity intrusions. While Generative Adversarial Networks (GANs) offer a solution for minority class synthesis, standard monolithic architectures suffer from majority-class gradient dominance, inevitably leading to conditional mode collapse.

To address this, we propose the Adaptive Manifold-Decoupled GAN (AMD-GAN), a novel framework that applies class-wise architectural decoupling to feature manifolds and dynamically adapts its optimization regime to each category's cardinality. This decoupled design mitigates inter-class interference and empirically reduces critic memorization. Evaluated across three heterogeneous NIDS benchmarks (CIC-IDS2017, UNSW-NB15, Edge-IIoTset 2022), AMD-GAN achieves a mean global empirical Wasserstein distance of $\mathcal{W}_1 = 0.009 \pm 0.001$ (across five independent seeds), confirming robust distributional fidelity. Under a Train-Synthetic-Test-Real (TSTR) protocol, balanced synthetic augmentation consistently improves Multiclass Macro F1-Scores. Specifically, in CIC-IDS2017, it yields a 10.1 pp absolute improvement (+45.2% Botnet, +22.9% Web Attack), demonstrating statistically significant superiority over interpolation methods like ADASYN and Borderline-SMOTE.

An extensive ablation study isolates the adaptive engine as the causal driver of manifold recovery, while Distance to Closest Record (DCR) analysis confirms the absence of exact-match memorization. Furthermore, interpretability techniques (SHAP and LIME) validate that the generated flows preserve authentic protocol semantics, enabling unambiguous attribution for Security Operations Centers (SOC). Deployed offensively as an automated Red Teaming engine, AMD-GAN exposes structural fragility in operational tree ensembles, inducing up to 0.909 Macro F1-Score degradation under volumetric stress scenarios. Finally, the decoupled sequential training architecture restricts peak GPU VRAM to 1,575 MB while generating 90,000 synthetic flows in 15.7 seconds, establishing AMD-GAN as a computationally efficient, low-VRAM solution potentially suitable for resource-constrained deployments for robust data augmentation and adversarial NIDS auditing.

Keywords: Network Intrusion Detection, Generative Adversarial Networks, Class Imbalance, Synthetic Data, Adaptive Training, Explainability

1. Introduction

Advanced persistent threats, botnets, and zero-day exploits share a common characteristic: they are statistically rare. In modern heterogeneous networks spanning traditional enterprise IT infrastructure, industrial control systems, and resource-constrained Internet of Things (IoT) deployments, malicious events constitute a small fraction of the total traffic volume. This rarity is not a mere data engineering artifact, but a structural property that Machine Learning (ML)-based Network Intrusion Detection Systems (NIDS) systematically fail to account for [1]. As these systems increasingly serve as the primary defensive layer against high-velocity and encrypted attack campaigns [2], their algorithmic blind spot toward sparse intrusion categories represents an exploitable vulnerability in its own right.

Real-world network traffic follows a long-tail distribution in which benign sessions are overwhelmingly predominant across all network domains [1]. Standard ML classifiers integrated in NIDS optimize for global accuracy, which

*Corresponding author.
Email addresses: alara@uma.es (Antonio Lara-Gutierrez), mcgago@uma.es (Carmen Fernandez-Gago), onieva@uma.es (Jose A. Onieva)

drives them to overfit the majority class and suppress the gradient signal of rare attack categories. The operational consequences are severe: a classifier that achieves 99% global accuracy may simultaneously exhibit near-zero detection capability for Botnet infiltrations or Web Attack campaigns, precisely the categories an adversary would leverage to evade an operational NIDS. This issue is not class-agnostic as it penalizes the detection of the most dangerous and low-frequency intrusion vectors.

To mitigate this bias, traditional oversampling techniques such as the Synthetic Minority Over-sampling Technique (SMOTE) have been widely employed [3]. However, these methods rely on linear interpolation, an assumption that is fundamentally incompatible with the topology of real-world attack traffic. Across traditional enterprise networks, high-noise industrial environments, and IoT telemetry streams, attack vectors form disjoint, non-convex clusters in feature space. Linear interpolation between minority samples in these geometries forces the synthesis of instances in the empty space between clusters, generating invalid protocol states and boundary noise that actively degrades classifier precision rather than improving minority class recall [4].

In contrast, Generative Adversarial Networks (GANs) offer an alternative by learning the underlying probability density function of the data rather than interpolating within it. However, standard Conditional GAN (cGAN) architectures are structurally ill-suited for severe class imbalance. In a shared framework, the discriminator is highly exposed to majority class samples and learns to associate "realness" almost exclusively with benign traffic features. This forces the generator to prioritize majority modes to minimize global loss, producing a failure known as conditional mode collapse: the synthesized minority samples either collapse into invalid protocol states or converge to a generalized approximation of the majority class [5, 6]. From a security perspective, this means that the very attack categories most critical to detect, like zero-day exploits or low-frequency intrusion vectors, are precisely those a monolithic generative model is least capable of synthesizing faithfully. Consequently, current state-of-the-art approaches fail to simultaneously achieve high statistical fidelity for sparse attack manifolds, dynamically adapt to extreme cardinality disparities, and maintain the computational scalability required for resource-constrained NIDS deployments.

To bridge this operational gap, we propose the Adaptive Manifold-Decoupled GAN (AMD-GAN), a novel generative framework designed to serve as both a defensive hardening tool and an offensive Red Teaming engine for operational NIDS. Rather than applying a uniform "one-size-fits-all" optimization regime, AMD-GAN deploys an ensemble of class-specific generator-discriminator pairs, each trained in architectural isolation on a single traffic category. This manifold decoupling mitigates the gradient dominance of majority traffic from suppressing the learning of minority attack features, enabling high-fidelity synthesis across all cardinality regimes. The same generative repository that hardens classifiers through balanced augmentation can be reconfigured to generate adversarial volumetric stress tests, exposing the structural fragility of deployed intrusion detectors before an attacker does.

The main contributions of this work span two operational pillars: *defensive* data augmentation for robust NIDS training, and *offensive* adversarial auditing to expose classifier vulnerabilities before real-world exploitation. Specifically:

- We introduce a decomposed generative architecture that isolates the optimization landscapes of different attack types and a novel heuristic strategy that dynamically adjusts the training physics based on the cardinality and density scaling of each one. This ensures that minority class generators are not penalized for failing to mimic majority class features, significantly mitigating the conditional mode collapse inherent in standard monolithic models.

- We operationalize AMD-GAN not merely as a data augmentation technique, but as a configurable adversarial benchmarking framework. By accepting user-defined distribution vectors, the synthesis engine can generate balanced datasets for defensive classifier training, and automated Red Teaming stress tests to systematically audit NIDS degradation under adversarial distribution shifts.

- We extensively validate the AMD-GAN framework across three diverse network intrusion benchmarks (CIC-IDS2017 [7], UNSW-NB15 [8], and Edge-IIoTset 2022 [9]) using a dual experimental protocol: assessing generative utility via Train-Synthetic-Test-Real (TSTR) classification across eight Machine Learning and Deep Learning (ML/DL) algorithms, and benchmarking its imbalance correction efficacy against state-of-the-art oversampling techniques.

- We conduct an exhaustive robustness evaluation, incorporating intrinsic statistical metrics to quantitatively demonstrate strong distributional alignment and a computational load analysis between our proposal and baseline models.

- We integrate an eXplainable AI (XAI) framework to validate the operational fidelity of our model. Through SHapley Additive exPlanations (SHAP) [10] and Local Interpretable Model-agnostic Explanations (LIME) [11], we demonstrate that AMD-GAN prevents feature contamination, captures genuine protocol-level semantics, and ensures unambiguous traceability.

The rest of this article is organized as follows: Section 2 critically reviews existing literature regarding class imbalance and monolithic GAN limitations. Section 3 formalizes the AMD-GAN framework, detailing its decoupled architecture and size-dependent adaptive training engine. Section 4 outlines the three heterogeneous NIDS benchmarks and the multifaceted evaluation protocols. Section 5 analyzes the experimental results, covering intrinsic statistical fidelity, downstream TSTR classification utility, oversampling efficacy, adversarial Red Teaming, and hardware profiling. Section 6 validates the framework's operational transparency and manifold separability using eXplainable AI techniques. Section 7 provides an empirical comparison against recent state-of-the-art generative models. Section 8 discusses the current constraints of the framework, and finally, Section 9 summarizes the primary contributions and directions for future work.

2. Related Work

Despite significant recent advances in generative modelling for network intrusion detection, the 2024–2026 state-of-the-art remains structurally incapable of simultaneously satisfying the three operational demands of an operational NIDS: strict distributional fidelity for rare attack manifolds, scalable inference latency compatible with high-speed traffic, and adaptive training dynamics across extreme cardinality disparities. We organize the critical analysis of existing approaches into three domains, covering hybrid rebalancing strategies, monolithic conditional architectures, and high-complexity specialized models, and demonstrating in each case why the identified gap remains unresolved. An overview is provided in Table 1.

Hybrid Rebalancing Strategies and Their Generative Limitations Early hybrid models such as TACGAN, proposed by Ding et al. (2022) [12] and IGAN-IDS, by Huang and Lei (2020) [13] established the foundational limitation of this category: combining classical undersampling with auxiliary GANs discards majority-class information and fails to resolve the underlying adversarial optimization conflict between class probability densities.

More recent 2025 proposals attempt structural hybridization but reproduce the same root failure. Menssouri and Amhoud (2025) [15] proposed CTGSM-DNN, a two-stage pipeline synthesizing rare IoT attacks via CTGAN before applying SMOTE with Edited Nearest Neighbors (SMOTE-ENN) to clean boundary noise. The requirement for a post-generation heuristic filter directly exposes the core weakness: the base generator cannot autonomously learn the exact topological manifold of rare attacks, and external filtering cannot compensate for structural mode collapse in the generative phase. From a security standpoint, this means that the rarest and most dangerous intrusion vectors are precisely those the generator fails to model faithfully.

Yang et al. (2025) [14] introduced CE-GAN, incorporating an encoder-decoder structure to compress high-dimensional noisy features. While dimensionality reduction stabilizes training, it simultaneously risks discarding the fine-grained protocol attributes that distinguish, for instance, a valid port scan from a reconnaissance precursor to a zero-day exploit. Kang et al. (2025) [16] designed M2M-VAEGAN, fusing a Variational Autoencoder (VAE) with a GAN and imposing continuous Gaussian priors via a Variational Gaussian Mixture model. However, forcing continuous distributional assumptions onto network protocols that are intrinsically discrete and disjoint artificially smooths the precise topological boundaries required to classify highly specific low-frequency threats. The common failure across the new proposals is architectural: all rely on unified generative channels that remain structurally exposed to majority-class gradient dominance, regardless of the sophistication of the surrounding pipeline.

Monolithic Conditional Architectures and Gradient Dominance The predominant approach in current literature is the use of monolithic Conditional GANs, where a single generator-discriminator pair attempts to learn the distribution of all classes simultaneously via label conditioning. Wang et al. (2022) presented CTTGAN [17], adapting the architecture to handle discrete and continuous variables in tabular traffic. Similarly, recent works by Lee and Park (2019) [19] and Rahman et al. (2025) [18] employ standard GANs to augment training sets for classifiers like Random Forest.

However, these shared architectures suffer from a biased gradient dominance. By exposing the discriminator overwhelmingly to benign traffic (often with ratios exceeding 10,000:1), the model optimizes its loss function based almost exclusively on majority features.

The practical security consequence is measurable: as demonstrated in our experimental evaluation (Section 5), a monolithic cGAN trained on CIC-IDS2017 produces a Wasserstein distance exceeding 0.284 globally and severely

Table 1: Comparison of AMD-GAN against state-of-the-art generative NIDS models based on architectural and operational features.

Reference	**Tabular Focus**	**No Undersampling**	**Decoupled Arch.**	**Protocol Adherence**	**Size-Adaptive Training**	**Edge-Compatible**	**No Post-Filter**	**Red Teaming**
Hybrid Rebalancing Strategies								
Ding et al. (2022) [12]	•			(P)		•	•	
Huang & Lei (2020) [13]	•			(P)		•	•	
Yang et al. (2025) [CE-GAN] [14]	•	•		(P)			•	
Menssouri & Amhoud (2025) [15]	•	•		(P)		•		
Kang et al. (2025) [M2M-VAEGAN] [16]	•	•					•	
Monolithic Conditional Architectures								
Wang et al. (2022) [17]	•	•		(P)		•	•	
Rahman et al. (2025) [18]	•	•		(P)		•	•	
Lee & Park (2019) [19]	•	•		(P)		•	•	
Ring et al. (2019) [20]	•	•				•	•	
High-Complexity Generative Models								
Yoon et al. (2019) [21]		•		•			•	
Cheng (2019) [PAC-GAN] [22]		•		•			•	
Schoen et al. (2024) [23]	•	•	N/A	•		•	•	
Ding et al. (2024) [24]	•	•	•	•		(P)	•	
Salehiyan et al. (2025) [25]	•	•		•			•	
Loodaricheh et al. (2026) [26]		•		•			•	
Awasthi et al. (2026) [27]		•		(P)		•	•	
AMD-GAN (Ours)	•	•	•	•	•	•	•	•

• = Supported; Blank = Not Supported; (P) = Partial/Limited support; N/A = Not Applicable.

diverges ($\mathcal{W}_1 > 0.10$) on critical minority classes, rendering its synthetic output practically useless for hardening a NIDS against the attack categories most likely to be exploited.

This structural bias forces the generator to prioritize majority modes to minimize global error, resulting in "conditional mode collapse." In this failure state, generated rare attacks either collapse into invalid protocol states or severely lack statistical diversity. Ring et al. (2019) [20] attempted to mitigate this representation issue by transforming discrete categorical attributes (IP addresses and ports) into continuous vector spaces using semantic embeddings (*IP2Vec*) and binary representations. While this continuous preprocessing stabilizes the generator and prevents the creation of malformed categorical features, it only serves as a data-level patch. As the authors acknowledge, the embedding translation restricts the generator to previously seen categorical values, limiting the topological diversity of novel synthetic attacks. The generator remains vulnerable to the gradient dominance of the majority class, failing to accurately model the topological distribution of sparse minority events.

High-Complexity Generative Models and Operational Scalability Constraints To overcome the limitations of standard cGANs, research has proposed increasingly complex architectures. Early specialized models like TimeGAN, proposed by Yoon et al. (2019) [21] attempted to capture the temporal dynamics of data sequences, while PAC-GAN, by Cheng (2019) [22] targeted packet-level generation. However, modelling traffic as raw sequential time-series imposes prohibitive computational costs compared to the flow-based statistics used in modern high-speed NIDS. Conversely, Schoen et al. (2024) [23] criticized the ability of generic GANs to maintain protocol adherence, suggesting a return to Bayesian Networks.

The most architecturally relevant prior work is TMG-GAN by Ding et al. (2024) [24], which successfully introduces a multi-generator structure to decouple the generation process and mitigate inter-class interference on CIC-IDS2017 and UNSW-NB15. However, TMG-GAN applies a *uniform optimization regime* across all decoupled generators, treating a majority-class manifold identically to a minority-class one. This architectural oversight is not a marginal inefficiency: without cardinality-aware hyperparameter adaptation, the shared discriminator in sparse manifolds rapidly memorizes the few available real samples rather than learning their underlying distribution, producing generators that achieve gradient stability at the cost of topological diversity.

More recently, the state-of-the-art (2025–2026) has shifted toward deep representation learning and multimodal synthesis. Salehiyan et al. (2025) [25] proposed an optimized Transformer-GAN-AE framework for Industrial IoT environments. While the transformer's self-attention captures intricate long-range dependencies, its quadratic $O(N^2)$ complexity poses severe scalability barriers for real-time edge inference. Similarly, Loodaricheh et al. (2026) [26] introduced MAGE-ID, shifting the paradigm toward multimodal diffusion models. Although MAGE-ID achieves unprecedented topological realism, the iterative Markov chain denoising required for generation introduces a prohibitive computational latency that is incompatible with the instantaneous volumetric synthesis demanded by modern NIDS. Concretely, a Red Teaming engine based on MAGE-ID's iterative denoising pipeline cannot generate adversarial volumetric floods at the throughput required to stress-test a production NIDS in real time, fundamentally limiting its utility as an offensive auditing tool.

Alternatively, Awasthi et al. (2026) [27] explored spatial transformations with TI-CNN, converting massive tabular flows into 2D image matrices for Convolutional Neural Network (CNN)-based synthesis, though such spatial encoding inherently distorts the discrete, categorical logic of network protocols. The resulting synthetic flows risk encoding structurally impossible protocol states, like invalid flag combinations or out-of-range port assignments, that would be filtered by a real packet inspector, producing a generative model that appears statistically plausible but is operationally invalid for NIDS augmentation.

The preceding analysis reveals that the current state-of-the-art remains constrained by uniform optimization in sparse manifolds, prohibitive synthesis latency, and protocol logic distortion. AMD-GAN is specifically designed to overcome these three critical failure modes by introducing an adaptive, decoupled architecture that ensures both mathematical fidelity and operational scalability.

3. Proposed Methodology

The proposed AMD-GAN framework addresses two distinct failure modes that jointly undermine the security utility of standard generative architectures for NIDS augmentation. The first is *class leakage*: when a shared generator learns all traffic categories simultaneously, the overwhelming statistical dominance of benign traffic causes synthetic attack samples to inherit majority-class feature distributions, producing flows that are statistically smooth but, in practice, indistinguishable from benign traffic, and therefore useless for hardening an intrusion detector. The second is *sparse manifold collapse*: for rare attack categories with fewer than a few thousand samples, a standard discriminator

memorizes the available real instances rather than learning their underlying distribution, causing the generator to converge to a degenerate low-diversity output that cannot capture the protocol-level diversity of real intrusion campaigns. AMD-GAN addresses both failure modes simultaneously by architecturally decoupling the feature space into class-specific manifolds and dynamically adapting the training hyperparameters and regularization to the exact cardinality of each class, ensuring that neither majority-class gradient dominance nor critic memorization can compromise the fidelity of synthesized attack traffic.

The framework is structured into three distinct phases: (A) Data Partitioning and Preprocessing, (B) Size-Dependent Adaptive Training, and (C) Configurable Synthesis. The overall architecture is illustrated in Figure 1.

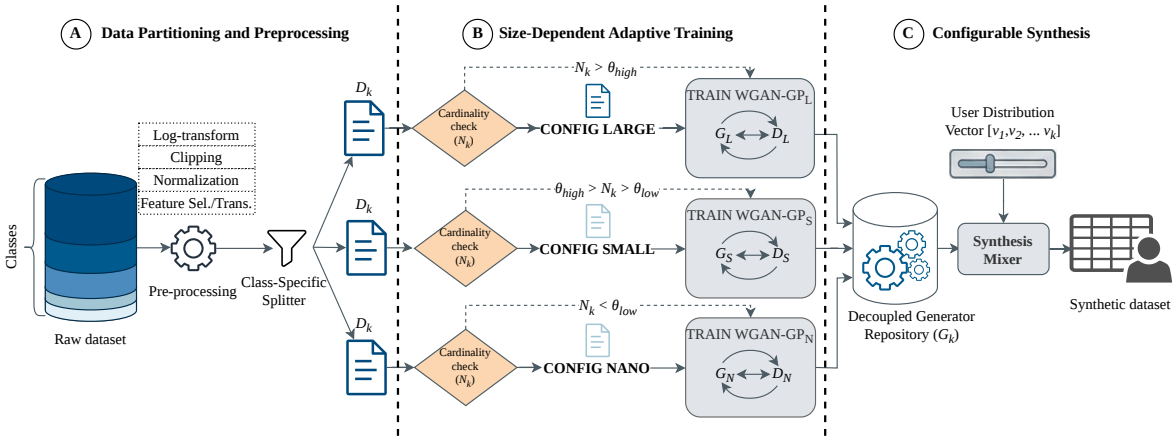

Figure 1: Proposed AMD-GAN architecture. **Phase A** decouples the imbalanced dataset into isolated manifolds. **Phase B** applies a size-dependent training engine, dynamically adjusting hyperparameters (λ, batch size, σ) based on class scarcity. **Phase C** enables configurable synthesis of balanced datasets or adversarial Red Teaming floods.

3.1. Manifold Decoupling and Preprocessing

Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ be the original imbalanced dataset, where $x_i \in \mathbb{R}^d$ represents the d-dimensional feature vector of a given network flow, and $y_i \in C = \{c_1, \ldots, c_k\}$ denotes its corresponding class label from a predefined set of k distinct traffic categories. Standard conditional GANs (cGANs) attempt to model the underlying data distribution $P(x|y)$ (the probability density of generating a feature vector x given a specific target class y) using a single, shared generator network $G(z|y)$. Here, $z \in \mathbb{R}^{d_z}$ represents a random latent noise vector sampled from a standard prior distribution, typically Gaussian ($z \sim \mathcal{N}(0, I)$). However, in NIDS scenarios, the majority class $c_{maj} \in C$ often constitutes over 80% of the data, dominating the gradient descent direction.

This imbalance leads to two compounding failure modes with direct security consequences. The first is *conditional mode collapse*: the generator maps the diverse latent space $\mathcal{Z}$ to a limited subset of outputs, effectively ignoring the conditional label y for minority classes. Mathematically, since the shared discriminator D minimizes a global loss expectation $\mathbb{E}_x$, the gradient contribution from the minority class, $\nabla_\theta \mathcal{L}_{min}$, becomes negligible compared to the majority class, $\nabla_\theta \mathcal{L}_{maj}$. The second is *class leakage*: $G(z|C_{min})$ fails to capture the true topology of the attack manifold and instead converges toward a generalized approximation of the majority class distribution. The resulting synthetic attack samples inherit benign traffic feature statistics; exhibiting, for instance, byte counts, inter-arrival times, or flag distributions characteristic of normal sessions rather than intrusion campaigns. A classifier trained on such poisoned synthetic data learns a decision boundary that is biased against detecting the very attacks the augmentation was intended to model. Label conditioning alone cannot resolve this: as long as the discriminator operates on a shared feature space dominated by benign traffic, the generator's gradient signal will remain corrupted regardless of the conditioning mechanism employed.

To resolve this, we apply class-wise architectural decoupling to partition $\mathcal{D}$ into K mutually disjoint subsets, mathematically defined as $\mathcal{D}_k = \{(x_i, y_i) \in \mathcal{D} \mid y_i = c_k\}$. This partitioning is strictly disjoint ($\mathcal{D}_i \cap \mathcal{D}_j = \emptyset$ for $i \neq j$) because each network flow in our formulation is mapped to a single, mutually exclusive category. It is important to note that this isolation is strictly achieved through class-wise algorithmic partitioning of the labeled dataset into mutually exclusive subsets, rather than through any physical or hardware-level separation of network environments. For each class-specific

manifold, we apply a dedicated preprocessing pipeline tailored for tabular network traffic. The pipeline consists of the following steps:

1. IP Address Expansion: We expand each IPv4 address into four independent continuous numerical features (octets). While IP addresses do not exhibit perfect continuous distance relationships, representing octets as continuous variables normalized via MinMax scaling is a calculated design choice. It allows the generator to approximate hierarchical subnet groupings (e.g., identifying class A or C network behaviors) without the prohibitive dimensionality explosion that One-Hot Encoding would introduce in sparse manifolds. Furthermore, as discussed in Section 2, alternative approaches like semantic embeddings (e.g., *IP2Vec*) strictly restrict the generator to previously observed IP values. In contrast, the continuous representation grants the GAN the necessary topological freedom to synthesize novel, previously unseen adversarial routing paths. The strict discrete integer nature of the octets (bounded to $[0, 255]$) is subsequently restored during the synthesis phase via heuristic rounding (see Section 3.3).

2. Logarithmic Smoothing: Network traffic data frequently contains continuous scalar features with heavy-tailed, non-negative empirical distributions (total transferred bytes or flow duration). To stabilize the adversarial gradient flow and prevent exponential divergence during training, these specific scalar components, denoted here as $x^{(j)}$, representing the j-th feature of the vector x_i, undergo a bounded logarithmic transformation: $x^{(j)}_{new} = \ln(1 + \max(0, x^{(j)}))$. The $\max(0, \cdot)$ operator acts as a strict non-negative rectifier, while the $+1$ shift is crucial to prevent asymptotic singularities (avoiding $\ln(0) \to -\infty$) when a flow feature is exactly zero. Finally, the outputs are clamped to a robust operating range of $[-20, 20]$, and any residual infinite or NaN anomalies are zeroed out.

3. MinMax Scaling: Since the generator's output layer uses a tanh activation function, all features are uniformly normalized to the $[-1, 1]$ range using a MinMax Scaler.

3.2. Size-Dependent Adaptive Training Engine

The core innovation of AMD-GAN is the *Adaptive Training Engine*. Architectural manifold decoupling resolves class leakage but introduces a new challenge: each isolated generator now faces the full difficulty of its class cardinality without the statistical support of the broader dataset. For majority-class manifolds, dense sample coverage provides sufficient gradient signal for stable convergence. For sparse minority attack manifolds, however, applying the same optimization physics leads to severe overfitting. In high-dimensional network traffic feature spaces, a discriminator trained on fewer than θ_{low} samples cannot distinguish between learning the underlying attack distribution and memorizing the specific real instances available. The result is a discriminator that overfits to sharp, narrow decision boundaries around the observed samples, providing the generator with gradient feedback that reflects the memorized training points rather than the true topological manifold of the attack category. Critically, this failure is invisible to standard loss metrics: the WGAN-GP loss can converge smoothly while the generator produces a degenerate approximation of the real attack manifold without observable loss degradation. The Adaptive Training Engine prevents this by inspecting the cardinality $N_k = |\mathcal{D}_k|$ of each class manifold and dynamically assigning it to one of three optimization regimes.

To ensure generalizability across arbitrary network environments, the regime boundaries are parameterized on the feature space dimensionality d. While a neural network's capacity to memorize instances is fundamentally governed by its total parameter count, our architecture dictates that the width of the input and early hidden layers scales proportionally with d (as detailed in Table 3). Consequently, scaling the threshold via d implicitly scales with the network's operational capacity. According to statistical learning theory [28], robust estimation of a manifold's covariance structure without overfitting requires the sample size to scale with $O(d^2)$. Consequently, we define:

$$\theta_{high} = \lceil \rho_{high} \cdot d^2 \rceil, \quad \theta_{low} = \lceil \rho_{low} \cdot d^2 \rceil \tag{1}$$

where $\lceil \cdot \rceil$ denotes the ceiling function, ensuring the thresholds resolve to discrete sample counts, and ρ_{high} and ρ_{low} act as density scaling coefficients. We define θ_{high} to identify dense manifolds where sample sufficiency allows reliable moment estimation via standard gradients, and θ_{low} to delineate the critical sparsity zone.

To empirically validate the $O(d^2)$ scaling and the selected ρ coefficients, a preliminary hyperparameter grid search was conducted on the training partition of the CIC-IDS2017 dataset during the initial framework design. By monitoring the critic memorization rates (indicated by rapid Wasserstein distance collapse) across different sparsity levels, the optimal scalar bounds were empirically established at $\rho_{low} \approx 0.8$ and $\rho_{high} \approx 3.5$. Crucially, once derived, these coefficients were fixed *a priori* and kept strictly constant across all subsequent experiments and across all three heterogeneous datasets evaluated in this study. They were not manually retuned per dataset, ensuring the engine acts as a generalizable heuristic rather than an overfitted mechanism. In this preliminary ablation study, we compared the proposed adaptive strategy against three alternative settings:

- No Adaptive Regime: Applying uniform optimization caused the Critic to memorize real samples in sparse manifolds, leading to severe conditional mode collapse and synthetic data with near-zero intra-class variance.
- Fixed Thresholds: Utilizing hardcoded numerical thresholds (e.g., $\theta_{low} = 5000$) proved highly fragile across heterogeneous networks. A 5,000-sample threshold provides adequate density for a 78-dimensional space like CIC-IDS2017, but it over-regularizes a lower-dimensional space like UNSW-NB15 ($d = 29$), slowing convergence.
- Linear $O(d)$ Scaling: Scaling thresholds linearly with feature dimensions severely underestimated the required sample density for robust manifold estimation in complex datasets (Edge-IIoTset), triggering premature overfitting.

Therefore, parameterizing the transition boundaries strictly on $O(d^2)$ with $\rho_{low} \approx 0.8$ and $\rho_{high} \approx 3.5$ provides the a generalizable scaling law capable of adapting the training physics across entirely distinct network topologies without manual hyperparameter retuning.

To synthesize high-fidelity tabular network traffic, AMD-GAN utilizes the Wasserstein GAN with Gradient Penalty (WGAN-GP) architecture [29]. Standard GANs optimize the Jensen-Shannon divergence, which notoriously leads to vanishing gradients and mode collapse when the real and generated distributions are disjoint. WGAN-GP resolves this limitation by minimizing the Wasserstein distance (Earth Mover's distance), which provides a continuous, meaningful gradient for the generator even when synthetic samples are initialized far from the real attack data. This mathematical property is what enables our decoupled generators to converge on rare attack categories without collapsing.

The framework consists of two adversarial neural networks: the Generator (G) and the Critic (D).

3.2.1. The Generator (G)

The Generator is a mapping function that takes a latent vector $z \in \mathbb{R}^{d_z}$ sampled from a standard normal distribution and maps it to the data space $\mathcal{X} \subset \mathbb{R}^d$. For tabular network traffic, G is designed as a Multi-Layer Perceptron (MLP). The final layer employs a hyperbolic tangent activation function (tanh) to ensure that the generated flow features, denoted as $\tilde{x} = G(z)$, are bounded within $[-1, 1]$, matching the preprocessed real data range. The objective of G is to minimize the Critic's score for these generated samples, defined by the loss function:

$$\mathcal{L}_G = -\mathbb{E}_{\tilde{x}\sim P_g}[D(\tilde{x})] \tag{2}$$

3.2.2. The Critic (D)

Unlike a standard discriminator that outputs a probability $P(y = \text{real}|x)$ via a sigmoid function, the Critic $D : \mathcal{X} \to \mathbb{R}$ outputs an unconstrained scalar score representing the "realness" of the input. The Critic is trained to approximate the Wasserstein distance $W(P_r, P_g)$ between the real data distribution P_r and the generated distribution P_g.

Because the direct computation of the Wasserstein distance is difficult in high dimensions, we rely on the Kantorovich-Rubinstein duality, which transforms the distance into an optimizable expectation:

$$\mathcal{W}_1(P_r, P_g) = \sup_{\|f\|_L \leq 1} \left(\mathbb{E}_{x\sim P_r}[f(x)] - \mathbb{E}_{\tilde{x}\sim P_g}[f(\tilde{x})]\right) \tag{3}$$

where $\|D\|_L \leq 1$ dictates that the supremum is taken over the set of all 1-Lipschitz continuous functions f. Intuitively, the 1-Lipschitz constraint ensures that the gradient norm of the Critic never exceeds 1 ($\|\nabla_x D(x)\|_2 \leq 1$). This mathematical bound forces the Critic's decision boundary to remain smooth and stable, preventing the abrupt gradient explosions that typically cause mode collapse in standard GANs.

3.2.3. Gradient Penalty (GP) Mechanism

To enforce the 1-Lipschitz constraint ($\|\nabla_x D(x)\|_2 \leq 1$) required for the validity of the Wasserstein metric, we reject weight clipping (used in early WGANs) as it leads to pathological behaviour in deep networks [30]. Instead, we enforce a soft constraint by adding a Gradient Penalty term to the loss function. We define an interpolated sample, $\hat{x}$, sampled uniformly along straight lines between pairs of real and generated points:

$$\hat{x} = \epsilon x + (1 - \epsilon)\tilde{x}, \quad x \sim P_r,\ \tilde{x} \sim P_g,\ \epsilon \sim \mathcal{U}[0, 1] \tag{4}$$

The Critic is then penalized if the gradient norm with respect to $\hat{x}$ deviates from 1. The objective function for the Critic is:

$$\mathcal{L}_D = \underbrace{\mathbb{E}_{\tilde{x}\sim P_g}[D(\tilde{x})] - \mathbb{E}_{x\sim P_r}[D(x)]}_{\text{Critic Loss}} + \underbrace{\lambda\, \mathbb{E}_{\hat{x}\sim P_{\hat{x}}}\left[(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2\right]}_{\text{Gradient Penalty}} \tag{5}$$

where λ is the penalty coefficient and $P_{\hat{x}}$ denotes the distribution of interpolated samples $\hat{x}$ as defined in Eq. 4. While standard implementations fix $\lambda = 10$ globally, we adopt an adaptive regularization strength depending on the manifold density. Specifically, in highly sparse manifolds (CONFIG SMALL and CONFIG NANO), the distance between isolated real data points is large. Without strong constraints, the Critic tends to overfit by creating excessively sharp decision boundaries around these few real samples. By increasing the penalty coefficient to $\lambda = 15.0$ and $\lambda = 20.0$ respectively, we enforce a much stricter 1-Lipschitz constraint. This penalizes steep gradients, forcing the Critic's function to remain smooth across the empty spaces of the sparse manifold, which ensures the Generator receives stable, directional feedback instead of erratic noise. The complete training physics for each regime are detailed in Table 2, and the step-by-step learning process is formalized in Algorithm 1.

Table 2: Adaptive hyperparameter configuration based on class cardinality (N_k).

Parameter	**CONFIG LARGE**	**CONFIG SMALL**	**CONFIG NANO**
Cardinality (N_k)	$N_k \geq \theta_{high}$	$\theta_{high} > N_k \geq \theta_{low}$	$N_k < \theta_{low}$
Target Examples	Benign, DoS	PortScan, SQL Injection	Bot, Heartbleed
Training Physics			
Batch Size	128	32	16
Training Epochs	15,000	25,000	30,000
Learning Rate	$1 \cdot 10^{-4}$	$5 \cdot 10^{-5}$	$2 \cdot 10^{-5}$
Critic Updates (n_{crit})	5	3	2
Regularization			
Gradient Penalty (λ)	10.0	15.0	20.0
Noise Injection (σ)	0.0	0.02	0.03
Oversample Factor	1×	10×	20×
Shared Parameters			
Latent Dim (z)	100		
Optimizer	Adam ($\beta_1 = 0.0, \beta_2 = 0.9$)		

3.2.4. Architecture and preprocessing design per regime

To ensure stable generative convergence across extreme cardinality disparities, the framework dynamically tailors its neural network capacity, optimization physics and anti-memorization regularizations to the specific sample density of each isolated manifold.

High-Data Regime (CONFIG LARGE) For classes where $N_k \geq \theta_{high}$, the manifold is dense. We employ a configuration with substantial batch sizes (128) and higher learning rates ($1 \cdot 10^{-4}$). This allows the model to approximate the expectation $\mathbb{E}$ accurately over the diverse modes of the distribution without the need for additional regularization or synthetic noise.

Medium-Data Regime (CONFIG SMALL) For intermediate classes where $\theta_{high} > N_k \geq \theta_{low}$, the manifold exhibits partial sparsity. Training with standard physics risks the early onset of discriminator overfitting. Therefore, we introduce a moderate regularization strategy. We lower the batch size to 32, reduce the learning rate, and increase the gradient penalty to $\lambda = 15.0$. This balances the stable generation of new topological variations with the preservation of existing spatial boundaries.

Low-Data Regime (CONFIG NANO) For severe minority classes where $N_k < \theta_{low}$, the manifold is highly sparse and standard optimization physics are insufficient. A critic exposed to fewer than θ_{low} real samples will memorize them within the first training epochs, creating sharp, overfitted decision boundaries that provide the generator with gradient signals pointing toward the memorized instances rather than the broader topological manifold. The generator converges

Algorithm 1 AMD-GAN Size-Dependent Adaptive Training for Class C_k

Require: Class-specific manifold $\mathcal{D}_k$, feature dimension d, thresholds θ_{high}, θ_{low} (see Eq. 1)
Ensure: Converged class-specific generator G_k

$N_k \leftarrow |\mathcal{D}_k|$
▷ — Phase B.1: Determine training physics (Table 2) —
if $N_k \geq \theta_{high}$ **then**
  $(batch,\ epochs,\ n_{crit},\ \lambda,\ \sigma) \leftarrow$ CONFIGLARGE
else if $N_k \geq \theta_{low}$ **then**
  $(batch,\ epochs,\ n_{crit},\ \lambda,\ \sigma) \leftarrow$ CONFIGSMALL
else
  $(batch,\ epochs,\ n_{crit},\ \lambda,\ \sigma) \leftarrow$ CONFIGNANO
end if
▷ — Phase B.2: Manifold densification (sparse regimes only) —
if $N_k < \theta_{high}$ **then**
  $\mathcal{D}_k \leftarrow$ OVERSAMPLE($\mathcal{D}_k$, factor)
▷ factor = 10× (Small) or 20× (Nano); see Table 2
end if
▷ — Phase B.3: WGAN-GP training loop —
Initialise θ_G, θ_D randomly
for epoch = 1 **to** $epochs$ **do**
  for $t = 1$ **to** n_{crit} **do** ▷ Critic update
    Sample real batch $x \sim \mathcal{D}_k$
    if $\sigma > 0$ **then** ▷ Gaussian noise injection (Eq. 6); sparse regimes only
      $x \leftarrow x + \varepsilon, \quad \varepsilon \sim \mathcal{N}(\mathbf{0},\ \sigma^2\mathbf{I})$
    end if
    Sample $z \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$; $\tilde{x} \leftarrow G_{\theta_G}(z)$
    Sample $\varepsilon_\alpha \sim \mathcal{U}[0, 1]$; $\hat{x} \leftarrow \varepsilon_\alpha\, x + (1 - \varepsilon_\alpha)\, \tilde{x}$
    $\mathcal{L}_{GP} \leftarrow \mathbb{E}\Big[\big(\|\nabla_{\hat{x}} D_{\theta_D}(\hat{x})\|_2 - 1\big)^2\Big]$
    $\mathcal{L}_D \leftarrow \mathbb{E}[D_{\theta_D}(\tilde{x})] - \mathbb{E}[D_{\theta_D}(x)] + \lambda\, \mathcal{L}_{GP}$
    $\theta_D \leftarrow \theta_D - \nabla_{\theta_D} \mathcal{L}_D$ ▷ Adam: β_1=0.0, β_2=0.9, lr from Table 2
  end for
  Sample $z \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ ▷ Generator update
  $\mathcal{L}_G \leftarrow -\mathbb{E}[D_{\theta_D}(G_{\theta_G}(z))]$
  $\theta_G \leftarrow \theta_G - \nabla_{\theta_G} \mathcal{L}_G$
end for
return $G_k \leftarrow G_{\theta_G}$

OVERSAMPLE: Random instance replication (sampling with replacement) applied to densify sparse minority classes. It functions synergistically with the Critic's Gaussian noise injection (σ) to prevent exact-match memorization, ensuring the generation of novel topological variants instead of exact training copies.

to a low-diversity cluster of near-identical synthetic flows that reproduce surface-level statistical properties of the observed samples but fail to capture the protocol variation necessary for a NIDS to generalize its detection capability. From a defensive standpoint, augmenting a classifier with such degenerate synthetic data for Botnet or Heartbleed traffic provides no improvement in detection recall, and from an offensive standpoint, the resulting generator cannot produce the volumetric diversity required for a credible Red Teaming stress test.

To transition highly sparse manifolds from the critical sparsity zone into a learnable density, Phase B.2 of our Adaptive Training Engine applies a random oversampling multiplier prior to GAN optimization. While this ensures that the network receives a viable quantity of gradient updates per epoch, naively oversampling a sparse class inherently risks severe discriminator memorization. If left unregularized, the Critic would simply memorize the duplicated instances, leading the Generator to produce exact duplicates (exact-match mode collapse), thereby creating a severe privacy and membership inference vulnerability.

To prevent this privacy risk and ensure the generation of authentic diversity over the oversampled data, the CONFIG SMALL and CONFIG NANO regimes apply three simultaneous structural regularizations:

- **Micro-Batching**: We drastically reduce the batch size (down to 16 in CONFIG NANO). This introduces essential stochastic noise into the gradient estimation, preventing the optimizer from settling into sharp local minima associated with memorized duplicated samples.
- **Gaussian Noise Injection**: We inject additive noise to the real inputs of the Critic during training to mathematically smooth the decision manifold:

$$x_{\text{noisy}} = x + \varepsilon, \quad \varepsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}) \tag{6}$$

  where $\mathcal{N}(0, \sigma^2 I)$ denotes a multivariate Gaussian distribution with a zero mean vector and covariance matrix $\sigma^2 I$. The noise standard deviation σ scales inversely with class cardinality ($\sigma = 0.02$ for CONFIG SMALL, $\sigma = 0.03$ for CONFIG NANO). This forces the Critic to generalize geometric regions rather than memorizing exact point coordinates.
- **Reduced Critic Capacity**: The number of critic updates per generator step (n_{crit}) is proportionally reduced (down to 3 and 2) to prevent the Critic from becoming too strong too quickly compared to the generator in sparse feature spaces.

Table 3 details the specific layer stacks for G and Critic D under each adaptive configuration.

Table 3: Unified neural network architecture stack for Generators (G) and Critics (D) across all configurations.

Stage	**Common Ops** *(Gen / Critic)*	**CONFIG LARGE** *(Units G / D)*	**CONFIG SMALL** *(Units G / D)*	**CONFIG NANO** *(Units G / D)*
Input	-	$dim(z)$ / d	$dim(z)$ / d	$dim(z)$ / d
Hidden 1	LReLU†+BN / LReLU† *(Dense)*	256 / 512	128 / 256	64 / 128
Hidden 2	LReLU†+BN / LReLU† *(Dense)*	512 / 256	256 / 128	128 / 64
Hidden 3	LReLU† / LReLU† *(Dense)*	256 / 128	128 / 64	64 / 32
Output	Tanh / Linear *(Dense)*	d / 1	d / 1	d / 1

† LeakyReLU slope $\alpha = 0.2$. BN = Batch Normalization (Gen only).

3.3. Decoupled Repository and Configurable Synthesis

Upon convergence, each class-specific generator G_k is serialized into a *Decoupled Generator Repository*. This modularity allows for the incremental update of specific attack models without retraining the entire system.

For the final synthesis phase, the framework accepts a user-defined target distribution vector $V = [v_1, v_2, \ldots, v_K]$, where v_k represents the desired number of samples for class C_k. The synthesis engine executes parallel sampling:

$$\mathcal{D}_{\text{syn}} = \bigcup_{k=1}^{K} \left\{ G_k(z_j) \;\middle|\; z_j \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad j = 1, \ldots, v_k \right\} \tag{7}$$

Finally, the synthetic samples $\tilde{x}$ generated by the ensemble must be mapped back to the original operational feature space to reconstruct a fully compatible synthetic dataset $\mathcal{D}_{syn}$. To achieve this, the synthesis mixer applies the exact inverse of the preprocessing pipeline in reverse order:

1. **Inverse MinMax Scaling:** First, the inverse of the MinMax scaler is applied to project the generated values from the $[-1, 1]$ hyperbolic tangent activation range back to their respective unbounded continuous spaces.
2. **Exponential Reversion:** Next, the specific heavy-tailed volumetric features that originally underwent logarithmic smoothing are reverted using the exponential function minus one: $x^{(j)} = \exp(\tilde{x}^{(j)}) - 1$.
3. **Heuristic Rounding and Bound Enforcement:** Since GANs output continuous numerical approximations, discrete network features must be structurally reconstituted. For any discrete feature index j, the continuous output $\tilde{x}^{(j)}$ is mapped to its valid operational domain using a rounding and clipping operation:

$$\tilde{x}_{disc}^{(j)} = \max\left(a_j, \min\left(b_j, \lfloor \tilde{x}^{(j)} \rceil\right)\right) \tag{8}$$

   where $\lfloor \cdot \rceil$ denotes rounding to the nearest integer, and $[a_j, b_j]$ represent the rigid protocol bounds for feature j. Under this formulation, outputs representing IPv4 octets are strictly bounded to $[0, 255]$, simulated network ports to $[1, 65535]$, protocol numbers to $[1, 255]$, and absolute integer counters are clipped to be non-negative ($a_j = 0$). This operation is applied exclusively to features designated as discrete in the dataset schema (ports, IP octets, flag counts, protocol numbers); continuous scalar features are retained in their real-valued post-exponential form.

This range-based heuristic enforces basic feature-domain constraints, ensuring that the generated features in the resulting $\mathcal{D}_{syn}$ are constrained within their valid numerical boundaries, significantly reducing out-of-range artifacts (e.g., negative packet counts or invalid port numbers).

4. Experimental Setup

To evaluate the proposed AMD-GAN framework across multiple dimensions of performance, utility, and efficiency, we designed a comprehensive evaluation methodology comprising seven distinct experimental protocols. First, we conduct an intrinsic mathematical analysis to quantify the exact distributional fidelity of the synthetic data. Second, we assess downstream classification utility via a Train-Synthetic-Test-Real (TSTR) protocol. Third, we benchmark the framework's imbalance correction efficacy against traditional oversampling algorithms. Fourth, we implement an adversarial volumetric stress test (Red Teaming) to audit NIDS degradation. Fifth, we evaluate potential memorization and privacy risks using a Distance to Closest Record (DCR) analysis. Sixth, we execute a comprehensive ablation study to isolate and validate the causal contribution of the individual Adaptive Training Engine components. Finally, we perform a hardware profiling and computational load analysis to guarantee deployment viability.

4.1. Datasets and Experimental Environment

To assess the generalization capabilities of the AMD-GAN framework across diverse network topologies, feature extraction paradigms, and threat landscapes, our experiments are conducted using three distinct NIDS benchmarks.

The first dataset, CIC-IDS2017 [7], serves as a standard benchmark representing traditional enterprise IT networks. Its features are extracted using the CICFlowMeter tool [31], emphasizing time-based statistical flow metrics. This dataset is characterized by extreme class imbalance, making it our primary baseline for evaluating the framework's behavior under severe cardinality disparities.

The UNSW-NB15 dataset [8] presents a distinct challenge. Unlike CIC-IDS2017, it utilizes the Bro and Argus tools for feature extraction, resulting in a fundamentally different topological space. It is characterized by severe intrinsic

noise and high inter-class overlapping. This dataset challenges the framework to map complex, non-linearly separable manifolds without suffering from mode collapse.

Finally, we incorporate the Edge-IIoTset 2022 [9], a modern dataset representing contemporary Internet of Things (IoT) and Edge computing architectures. This dataset introduces novel attack vectors and scales the classification problem to 15 distinct classes. Its inclusion serves to validate both the structural scalability of our decoupled architecture and its direct applicability to next-generation telemetry data.

To ensure full reproducibility of the generative framework and the downstream TSTR evaluations, the details of the exact feature engineering pipeline applied to each NIDS benchmark is detailed here. Rather than feeding raw, heterogeneous network captures directly into the GAN, a preprocessing protocol was enforced to ensure that the models learned underlying topological relationships rather than memorizing localized, non-generalizable artifacts. Table 4 illustrates the evolutionary trajectory of the feature sets—from their original raw state to their final mathematical representation injected into the AMD-GAN framework.

Table 4: Evolutionary pipeline of feature topologies from raw network captures to the final preprocessing state utilized for generative modeling across the evaluated benchmarks.

Dataset	Feature Group	Original State	Transformation Process	Final Representation
CIC-IDS2017 (Raw $d \approx 84$)	Identifiers & Time	`Flow ID, Timestamp`	Discarded to prevent temporal or sequential memorization.	Dropped (0 features)
	Spatial Routing	`Source IP, Destination IP`	String split and numeric coercion (IPv4 Octet Expansion).	`Src_IP_[1-4], Dst_IP_[1-4]` (8 features)
	Flow Topology	~78 statistical flow metrics	Filtered to core cross-compatible metrics. Log-smoothing applied to heavy-tailed features.	`Source Port, Destination Port, Protocol, Total Fwd Packets... PSH Flag Count` (21 features)
Final $d = 78$	*Retained Core (21): Src/Dst Port, Protocol, Fwd/Bwd Packets, Lengths, Flow Duration, IAT Means/Stds, Packet Lengths, Flags (SYN, ACK, FIN, RST, PSH).*			
UNSW-NB15 (Raw $d = 49$)	Metadata & Strings	`id, state, service, attack_cat`	Discarded categorical/string fields incompatible with continuous WGAN boundaries.	Dropped (0 features)
	Spatial Routing	`Src IP, Dst IP`	String split and numeric coercion (IPv4 Octet Expansion).	`Src_IP_[1-4], Dst_IP_[1-4]` (8 features)
	Packet Dynamics	~43 flow & packet metrics	Filtered to match the 21 core topological features of CIC-IDS. Log-smoothing applied.	`Total Fwd Packet, Flow IAT Mean, SYN Flag Count...` (21 features)
Final $d = 29$	*Retained Core (21): Matched identically to CIC-IDS2017 topology to ensure framework generalizability and structural consistency.*			
Edge-IIoTset (Raw $d = 61$)	Frames & Payloads	`frame.time, tcp.payload, mqtt.msg, mqtt.topic...`	Discarded unique timestamps, non-numerical text payloads, and redundant protocol string flags.	Dropped (0 features)
	Spatial Routing	`ip.src_host, ip.dst_host`	String split and numeric coercion (IPv4 Octet Expansion).	`Src_IP_[1-4], Dst_IP_[1-4]` (8 features)
	Protocol Headers	50 mixed protocol flags & metrics (ARP, TCP, MQTT...)	Retained dense IoT-specific protocol properties. Log-scaled bytes, sequences, and ports.	`arp.opcode, tcp.flags, mqtt.len, udp.time_delta...` (50 features)
Final $d = 58$	*Retained Core (50): arp.* (2), icmp.* (4), http.* (9), tcp.* (13), udp.* (3), dns.* (7), mqtt.* (9), mbtcp.* (3). All features normalized sequentially.*			

For each experiment, we will synthesize two distinct datasets to isolate and measure different performance phenomena:

- **Syn-Real:** A synthetic dataset engineered to mirror the exact imbalanced class distribution of the original network traffic. This dataset serves as a fidelity baseline for the Train on Synthetic, Test on Real (TSTR) protocol. By deliberately maintaining the original scarcity of minority attacks, we expect the downstream classifier to exhibit the same inherent majority bias as the baseline model. This isolates and validates the pure topological realism of the generated samples without the confounding effects of data augmentation.
- **Syn-Balanced:** A targeted augmentation dataset where the generative ensemble oversamples severe minority attacks up to a predefined threshold, ensuring uniform cardinality across all classes. This dataset is designed to evaluate the framework's imbalance correction capabilities. By mitigating the original gradient dominance of the majority class, we expect this dataset to yield a substantial improvement in the classifier's Macro F1-Score for critical minority vectors.

To ensure strict reproducibility across all experiments, all data splitting, shuffling, and model weight initializations were executed using a fixed random seed (42). Furthermore, the specific feature sets extracted for each benchmark,

comprising 78 features for CIC-IDS2017, 29 features for UNSW-NB15, and 58 features for Edge-IIoTset 2022 (see Appendix B for a complete list of features per dataset), alongside the exact cardinality counts are presented in Table 5. This details the exact composition of the training partitions before and after the synthetic augmentation process.

Table 5: Class distributions, generative targets for experimental evaluation (Syn-Real and Syn-Balanced), and isolated training times across evaluated datasets.

Dataset	Target Class	Regime	Original Train	Syn-Real	Syn-Balanced	GAN Training Time
CIC-IDS2017 (d = 78)	BENIGN	LARGE	2,272,895	2,272,895	280,000	~ 59 min
	DoS	LARGE	252,660	252,660	280,000	~ 59 min
	Port Scan	LARGE	158,930	158,930	280,000	~ 58 min
	DDoS	LARGE	128,027	128,027	280,000	~ 65 min
	Brute Force	SMALL	13,835	13,835	280,000	~ 58 min
	Web Attack	NANO	2,180	2,180	280,000	~ 59 min
	Bot	NANO	1,966	1,966	280,000	~ 60 min
UNSW-NB15 (d = 29)	Benign	LARGE	3,390,752	3,390,752	280,000	~ 64 min
	Exploits	LARGE	30,951	30,951	280,000	~ 60 min
	Fuzzers	LARGE	29,613	29,613	280,000	~ 60 min
	Reconnaissance	LARGE	16,735	16,735	280,000	~ 59 min
	Generic	LARGE	4,632	4,632	280,000	~ 60 min
	DoS	LARGE	4,467	4,467	280,000	~ 59 min
	Shellcode	NANO	2,102	2,102	280,000	~ 60 min
Edge-IIoTset (d = 58)	Normal	LARGE	1,615,643	1,615,643	130,000	~ 63 min
	DDoS_UDP	LARGE	121,568	121,568	130,000	~ 62 min
	DDoS_ICMP	LARGE	116,436	116,436	130,000	~ 63 min
	SQL_injection	LARGE	51,203	51,203	130,000	~ 63 min
	Password	LARGE	50,153	50,153	130,000	~ 63 min
	Vuln_scanner	LARGE	50,110	50,110	130,000	~ 71 min
	DDoS_TCP	LARGE	50,062	50,062	130,000	~ 67 min
	DDoS_HTTP	LARGE	49,911	49,911	130,000	~ 73 min
	Uploading	LARGE	37,634	37,634	130,000	~ 64 min
	Backdoor	LARGE	24,862	24,862	130,000	~ 63 min
	Port_Scanning	LARGE	22,564	22,564	130,000	~ 63 min
	XSS	LARGE	15,915	15,915	130,000	~ 64 min
	Ransomware	SMALL	10,925	10,925	130,000	~ 62 min
	MITM	NANO	1,214	1,214	130,000	~ 62 min
	Fingerprinting	NANO	1,001	1,001	130,000	~ 62 min

In order to contextualize the theoretical parameters established in Section 3.2, Table 6 maps the specific feature dimensionality (d) of each evaluated benchmark to its corresponding operational thresholds (θ_{low} and θ_{high}) calculated via the $O(d^2)$ scaling law. This dynamic calibration ensures that the framework evaluates class cardinality not as an absolute number, but relative to the topological complexity of the specific dataset, distributing the attack classes across the appropriate optimization regimes (Nano, Small, Large).

Table 6: Mapping of dataset dimensionality to adaptive $O(d^2)$ operational thresholds and regime distributions.

Dataset	Features (d)	Complexity (d^2)	θ_{low} ($\approx 0.8d^2$)	θ_{high} ($\approx 3.5d^2$)
CIC-IDS2017	78	6,084	4,868	21,294
Edge-IIoTset 2022	58	3,364	2,692	11,774
UNSW-NB15	29	841	673	2,944

To resolve any ambiguity regarding dataset class mappings and ensure full reproducibility, Table 7 explicitly details the class aggregation and exclusion criteria applied to the raw datasets.

It is critical to distinguish between the intrinsic generative evaluation and the downstream classification. For CIC-IDS2017, granular sub-categories were grouped by primary attack vector and extremely rare classes such as *Heartbleed* (11 samples) and *Infiltration* (36 samples) were retained exclusively for the intrinsic generative fidelity evaluation ($\mathcal{W}_1$ analysis in Figure 12) to demonstrate the AMD-GAN Nano configuration's capability to converge under extreme sparsity. However, they were purposely excluded from the downstream TSTR classification benchmarks (Table 5)

because a 30% test split would leave insufficient samples for a statistically significant machine learning evaluation. For UNSW-NB15, ambiguous minority categories (*Analysis*, *Backdoors*, *Worms*) were excluded due to insufficient samples for robust train/test splitting and high topological overlap, streamlining the evaluation to 7 distinct manifolds.

Table 7: Dataset class mapping, aggregation, and exclusion criteria.

Dataset	Classes (Orig → Final)	Features	Mapped / Retained Classes	Excluded / Rare Classes
CIC-IDS2017	15 → 7	78	**Benign**, **DoS** (Merged: Hulk, GoldenEye, Slowloris, Slowhttptest), **DDoS**, **Port Scan**, **Brute Force** (Merged: FTP, SSH), **Web Attack** (Merged: Brute Force, XSS, SQLi), **Bot**.	**Heartbleed**, **Infiltration** (Excluded from TSTR due to extreme sparsity; evaluated only for intrinsic GAN fidelity).
UNSW-NB15	10 → 7	29	**Benign**, **Exploits**, **Fuzzers**, **Reconnaissance**, **Generic**, **DoS**, **Shellcode**.	**Analysis**, **Backdoors**, **Worms** (Excluded due to insufficient cardinality and high overlap).
Edge-IIoTset	15 → 15	58	All 15 original categories retained (e.g., Normal, DDoS_UDP, SQL_injection, etc.).	None.

All experiments were executed on an AI-oriented server equipped with an Intel(R) Xeon(R) Gold 5418Y (48 physical cores, 96 logical cores), 503.4 GB of DDR memory, and 4 NVIDIA L40S GPUs (48 GB GDDR6 each) running Debian GNU/Linux 12. While this high-capacity infrastructure was utilized strictly to parallelize the extensive classification benchmarks and oversampling baselines across the three datasets, all computational load and hardware profiling metrics for AMD-GAN (Section 5.5) were measured by isolating a single GPU process.

4.2. Data Splitting Protocol and Leakage Prevention

To ensure the integrity of the generative evaluation and explicitly prevent any form of data leakage, a data partitioning and preprocessing protocol was enforced across all experimental benchmarks. Addressing this is particularly critical for the validity of the Train-Synthetic-Test-Real (TSTR) protocol. The methodology adhered to the following principles:

- Prior to partitioning, exact duplicate network flows were removed from the raw datasets. This step is important to prevent the GAN's Critic from memorizing duplicated instances, and to ensure downstream classifiers are not wrongly evaluated on identical samples present in their training set.
- The deduplicated datasets were split into a 70% training partition and a 30% hold-out test partition using random stratified sampling to maintain the precise long-tail cardinality ratio of the original network traffic. As detailed in Table 4, temporal identifiers (timestamps) and strict categorical host artifacts were explicitly discarded prior to this split to eliminate temporal correlation leakage.
- All data preprocessing mechanisms and feature transformations mentioned in Section 3.3 (including MinMax scalers and logarithmic bounds) were fitted exclusively on the 70% training partition. The test set was transformed using only these pre-fitted parameters, ensuring that no global statistical information leaked into the generative or classification phases.
- During the manifold decoupling phase (Section 3.2), each class-specific generator (G_k) and critic (D_k) within the AMD-GAN framework was trained using the isolated subsets of samples derived solely from the 70% training partition.
- The 30% test partition was entirely hidden during the GAN optimization and synthetic data generation processes. Furthermore, the volumetric targets for class balancing (the specific oversampling thresholds) were calculated strictly based on the cardinality of the training partition alone. Consequently, the generative repositories possess no representations of the test set, ensuring that the real test data remained completely unseen by both the generative algorithms and the downstream ML classifiers prior to final evaluation.

4.3. Intrinsic Statistical Protocols

Before testing the synthetic datasets on downstream ML classifiers, we aim to demonstrate that the generative model successfully replicates the true topological manifold of the network traffic. To evaluate this intrinsic fidelity across the three datasets, we employ two mathematical metrics:

Empirical Wasserstein Distance ($\mathcal{W}_1$): We compute the 1D Earth Mover's Distance feature by feature to measure marginal distribution similarity. Given two probability distributions P (real data) and Q (synthetic data), the empirical, $\mathcal{W}_1$, is defined using their inverse cumulative distribution functions, F^{-1}, as follows:

$$\mathcal{W}_1(P, Q) = \int_0^1 |F_P^{-1}(q) - F_Q^{-1}(q)|dq \tag{9}$$

where P denotes the real data distribution and Q the synthetic distribution. The metric is calculated over MinMax jointly normalized data within the range $[0, 1]$. We report aggregate statistics across all features. Lower $\mathcal{W}_1$ values indicate higher marginal similarity.

Multivariate Maximum Mean Discrepancy (MMD^2): While $\mathcal{W}_1$ captures 1D marginals, MMD^2 measures the divergence between multivariate distributions, effectively capturing complex, non-linear feature correlations. By implicitly mapping the distributions into a Reproducing Kernel Hilbert Space (RKHS), a high-dimensional mathematical space where non-linear data structures can be compared using linear distance metrics, MMD^2 evaluates the distance between their mean embeddings and is computed as:

$$MMD^2(P, Q) = \mathbb{E}_{x,x' \sim P}[k(x, x')] - 2\,\mathbb{E}_{\substack{x \sim P \\ y \sim Q}}[k(x, y)] + \mathbb{E}_{y,y' \sim Q}[k(y, y')] \tag{10}$$

We employ the Radial Basis Function (RBF) kernel, where the bandwidth parameter γ is dynamically estimated using the median heuristic of pairwise distances. A lower MMD^2 score denotes superior multivariate alignment.

4.4. Extrinsic Utility Protocols (TSTR & Oversampling)

To quantify the practical value of the generated data for cybersecurity applications, we evaluate its downstream classification utility and its efficacy for imbalance correction.

Experiment I: TSTR Methodology. This experiment evaluates whether the synthetic traffic captures the underlying statistical properties of real network flows sufficiently to train a competent NIDS. We adopt the TSTR methodology, comparing it against the Train-Real-Test-Real (TRTR) baseline, which serves as the upper bound of performance. We defined four specific evaluation scenarios:

1. **TSTR Multiclass & Binary:** The NIDS is trained exclusively on synthetic data to distinguish either between all specific attack classes (Multiclass) or simply Benign vs. Attack (Binary). The models are then evaluated on a hold-out test set of purely real network flows.
2. **TRTR Multiclass & Binary (Baselines):** The NIDS is trained and tested entirely on real data.

For rigorous validation, we employed a comprehensive suite of machine learning algorithms to ensure model-agnostic results. To guarantee full reproducibility, the core architectural hyperparameters were explicitly defined as follows: a *Dummy* baseline (most frequent strategy), *Logistic Regression* (SAGA solver, max iterations=1,000), *K-Nearest Neighbors* (KNN, $k = 5$), *Random Forest* ($n = 200$, balanced subsample class weighting), *LightGBM* ($n = 100$), *XGBoost* ($n = 100$), a Multi-Layer Perceptron (*MLP*, hidden layers: [128, 64], max iterations=500, with early stopping), and a *Soft Voting* Meta-Ensemble (comprising Random Forest, Extra Trees, and LightGBM estimators). The exact initialization parameters and fixed random seed configurations across all models are fully documented in the public code repository (see Data and Code Availability Section).

Experiment II: Imbalance Correction Benchmarking. This experiment positions AMD-GAN as a data augmentation technique. We compare our generative proposal against six established oversampling methods: Random Oversampling (ROS), SMOTE, Borderline-SMOTE, ADASYN, and the hybrid SMOTE-ENN algorithm. For this benchmark, we compare four distinct models to evaluate which augmentation technique maximizes detection performance on the minority classes without degrading precision on the majority class. Performance is primarily measured using the Macro F1-Score.

4.5. Robustness and Red Teaming Protocols

Unlike traditional evaluations where GANs are used to train classifiers, this protocol employs AMD-GAN as an offensive Red Teaming tool to audit the robustness of existing operational NIDS across all three heterogeneous benchmarks, once we have demonstrated the practical and realistic value of the generated data. Four standard classifiers (Random Forest, KNN, LightGBM, and Logistic Regression) are trained on the original, highly imbalanced real datasets. We then subject these "victim" models to four extreme, GAN-generated rare-class volumetric stress tests without retraining:

- **Scenario A (Balanced Baseline):** A controlled environment where Benign traffic is constrained to a fixed minority of the total volume, while the specific attack classes are artificially equalized.
- **Scenario B (Rare-Class Saturation):** A targeted volumetric saturation where synthetic flows of the rarest attack class overwhelmingly dominate the network against a marginal Benign baseline.
- **Scenario C (Volumetric Flood):** A denial-of-service simulation completely dominated by compound volumetric attack vectors, severely reducing Benign traffic.
- **Scenario D (Inverse Imbalance):** A critical environment where Benign traffic is reduced to a severe minority, while all malicious vectors collectively and evenly saturate the network.

This protocol measures the performance degradation (Δ Macro F1-Score) of traditional tree-based and linear algorithms when exposed to sudden, adversarial distributional shifts.

4.6. System Profiling

To justify the architectural design choices and ensure the framework's deployment viability in hardware-constrained environments, we conduct one final structural evaluation.

We perform real-time hardware profiling tracking GPU VRAM footprint, total trainable parameters, gradient operations, and inference latency (measured in seconds per 90,000 samples). AMD-GAN is benchmarked against two standard paradigms: a monolithic Conditional GAN (cGAN) and an unregularized decoupled GAN array. Generative sample quality is additionally compared against the two baseline architectures using the per-class $\mathcal{W}_1$ analysis reported in Section 5.5.

4.7. Memorization and Privacy Risk Protocols

To validate that AMD-GAN synthesizes novel attack variants rather than memorizing and replaying the oversampled sparse classes, we introduce a privacy and diversity evaluation protocol based on the Distance to Closest Record (DCR). Following standard evaluation frameworks for privacy-preserving synthetic data generation [32], we compute three primary metrics over the unprojected, scaled feature space to systematically audit memorization and membership inference risks:

1. **Exact Match Rate (Duplicate Rate):** The percentage of generated synthetic samples that possess a Euclidean distance of exactly zero to any sample in the real training dataset. A rate near 0% confirms the absence of pure mode collapse and exact-match memorization.
2. **Distance to Closest Record (DCR_{train} vs DCR_{test}):** We calculate the mean Euclidean distance from each synthetic sample to its absolute nearest neighbor in the generative training split (DCR_{train}) and the hold-out test split (DCR_{test}). A model that generalizes properly without overfitting should yield a DCR ratio (DCR_{train}/DCR_{test}) approaching 1.0.
3. **Membership Inference Privacy Risk:** Evaluated as the probability of a synthetic sample being strictly closer to the training set than to the unseen test set, defined as $P(DCR_{train} < DCR_{test})$. A secure, non-overfitted generative model should yield a privacy risk metric approaching 0.50 (50%), indicating that the synthetic data orbits the true underlying distribution.

5. Results and Analysis

In this section, we present an evaluation of the AMD-GAN framework, structured around a total of six experimental protocols defined in Section 4.

5.1. Intrinsic Statistical Fidelity Analysis

To validate the stability of the proposed Adaptive Training Engine, we first analyze the generative convergence and the quantitative statistical fidelity of the synthetic feature distributions. To visually and quantitatively assess this stability, Appendix A presents a comprehensive set of training loss curves and Kernel Density Estimation (KDE) diagrams across representative classes from our three evaluated datasets.

As detailed visually in Appendix A, the training dynamics vary depending on the cardinality regime. For the *High-Data Regime*, the majority class benefits from the high-capacity configuration. The training dynamics are characterized

by smooth loss curves and rapid convergence, allowing the generator to perfectly capture the dominant modes of the traffic without regularization. Conversely, the *Low-Data Regime* illustrates the critical challenge of learning from extreme sparsity (< 0.1% of the dataset). While standard GANs suffer from immediate discriminator overfitting here, our *Nano Configuration* intentionally induces a high-frequency oscillation in the loss curve. The stochastic noise introduced by micro-batching and heuristic noise injection prevents the Critic from memorizing the few available real samples.

This visual stability is quantitatively corroborated by our mathematical metrics. To ensure these claims are statistically grounded and not artifacts of a single sampling event, we computed the intrinsic distances across 5 independent generative seeds. Table 8 presents the aggregate $\mathcal{W}_1$ and MMD^2 across all datasets and synthesis strategies, reported as Mean ± Standard Deviation. Based on evaluation thresholds, a $\mathcal{W}_1 < 0.01$ denotes excellent fidelity (virtually identical marginal distributions), while $\mathcal{W}_1 < 0.05$ indicates good adherence.

Table 8: Intrinsic statistical fidelity ($\mathcal{W}_1$ and MMD^2) across evaluated datasets. Results are reported as Mean ± Standard Deviation over 5 independent generative seeds.

Synthesis Strategy	**Classes**	$\mathcal{W}_1$ **Distance**		MMD^2	
		Average	**Worst (Max)**	**Average**	**Worst (Max)**
Dataset: CIC-IDS2017					
Syn-Real	9	0.0092 ± 0.0008	0.0305 ± 0.0021	0.0128 ± 0.0014	0.0794 ± 0.0052
Syn-Balanced	9	0.0092 ± 0.0007	0.0305 ± 0.0019	0.0124 ± 0.0012	0.0761 ± 0.0048
Dataset: UNSW-NB15					
Syn-Real	7	0.0066 ± 0.0005	0.0134 ± 0.0011	0.0039 ± 0.0006	0.0119 ± 0.0014
Syn-Balanced	7	0.0066 ± 0.0006	0.0131 ± 0.0010	0.0041 ± 0.0005	0.0119 ± 0.0012
Dataset: Edge-IIoTset 2022					
Syn-Real	15	0.0093 ± 0.0010	0.0145 ± 0.0015	0.0267 ± 0.0022	0.0708 ± 0.0045
Syn-Balanced	15	0.0092 ± 0.0009	0.0141 ± 0.0013	0.0268 ± 0.0020	0.0704 ± 0.0041

Note: $\mathcal{W}_1 < 0.01$ indicates Excellent Fidelity while $\mathcal{W}_1 < 0.03$ indicates Good Fidelity; $MMD^2 < 0.05$ indicates Moderate to Excellent alignment.

The framework achieves an average $\mathcal{W}_1$ below 0.01 across all three datasets, confirming excellent generative fidelity across all evaluated benchmarks. Remarkably, the absolute worst-case $\mathcal{W}_1$ distance across all evaluated models is 0.0305 (in CIC-IDS2017). This indicates that even the most complex, heavily skewed attack manifolds strictly remain within the *Good* fidelity boundary, staying far below the 0.10 threshold that would indicate significant structural divergence. Similarly, the average MMD^2 scores consistently fall between 0.0039 and 0.0268, indicating that the WGAN-GP not only replicates 1D features but also reasonably preserves the complex multivariate correlations inherent to network traffic.

To provide a granular view of this performance across diverse cardinality levels, Figure 2 maps the joint quality assessment ($\mathcal{W}_1$ vs. MMD^2) for every individual attack class. The scaling of the dot size reveals that the generative fidelity is completely decoupled from the original class sparsity. Whether synthesizing massive manifolds like benign traffic (millions of samples) or extremely rare events like web attacks or fingerprinting ($\approx 1,000$ samples), all data points cluster exclusively within the *Excellent* and *Good* bounded regions.

5.2. Extrinsic Utility: TSTR Classification

Having established statistical evidence of fidelity, we evaluate the downstream utility of the synthetic data through the Train-Synthetic-Test-Real (TSTR) protocol. Classifiers are trained solely on generated traffic and evaluated on a held-out real test set. To establish the upper bound of achievable performance, we report the results for the best-performing model from our ensemble against the Train-Real-Test-Real (TRTR) baseline. To address the variance in GAN sampling and support the generality of our claims, all GAN-based TSTR experiments were executed across five independent random seeds. Results are reported as Mean ± Standard Deviation.

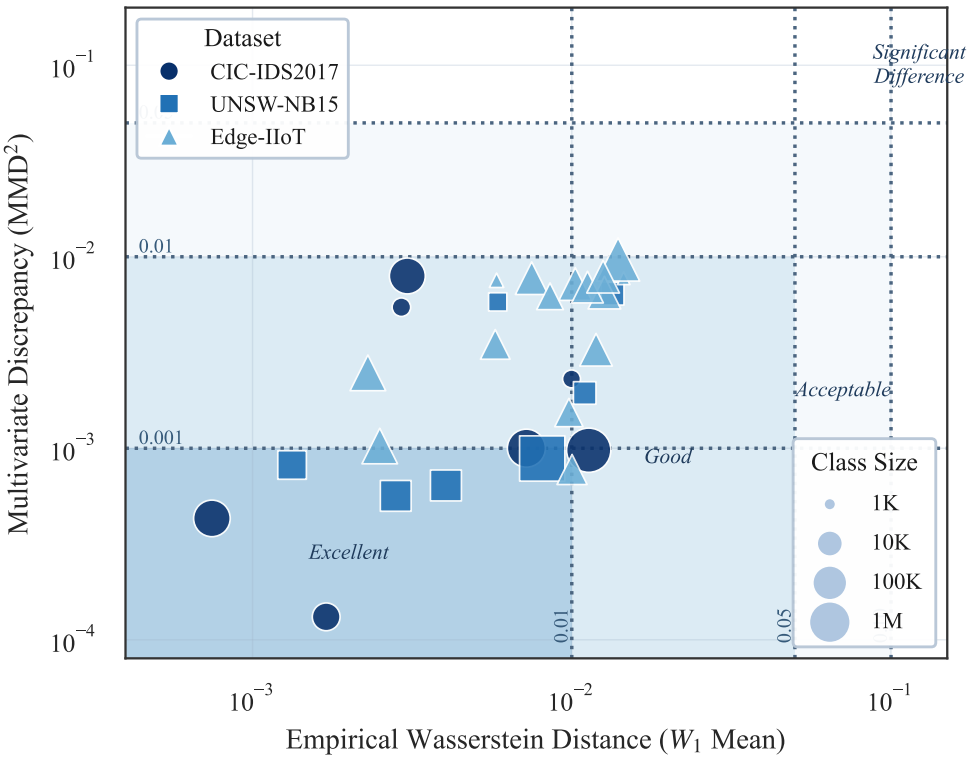

Figure 2: Joint quality assessment ($\mathcal{W}_1$ vs. MMD^2) of synthetic manifolds for each generated class across all benchmarks.

5.2.1. Binary Classification Fidelity

First, we assess the capability of AMD-GAN to preserve the fundamental decision boundary between benign and malicious traffic. Table 9 compares the performance across the three heterogeneous benchmarks.

Table 9: Binary classification performance (Benign vs. Attack) across all datasets (Mean ± Std. Dev).

Dataset	**TRTR (Upper Bound)**	**TSTR (Synthetic Ours)**		**Gap**
	Real Baseline	*Syn-Real*	*Syn-Balanced*	*(Syn-Bal. vs TRTR)*
CIC-IDS2017	**0.9994** ± 0.000	0.9834 ± 0.002	0.9987 ± 0.001	-0.0007
UNSW-NB15	**0.8931** ± 0.004	0.8783 ± 0.005	0.8877 ± 0.003	-0.0054
Edge-IIoTset	**0.9946** ± 0.002	0.9741 ± 0.004	0.9899 ± 0.003	-0.0047

In the binary context, the *Syn-Balanced* dataset consistently achieves Macro F1-Scores that virtually mirror the real baselines across all environments. The fidelity gap is practically negligible (e.g., $\Delta \leq 0.0007$ in CIC-IDS2017), confirming that AMD-GAN generates highly valid generalized attack signatures.

5.2.2. Multiclass Utility and Imbalance Correction

The multiclass scenario presents a far more rigorous challenge, requiring the model to distinguish between highly overlapping specific attack vectors across heterogeneous network topologies. Table 10 details this comprehensive comparison.

While the *Syn-Real* datasets achieve high global accuracy, their depressed Macro F1-Scores underscore the difficulty of learning minority manifolds from sparse synthetic data. However, the *Syn-Balanced* augmentation demonstrates the true operational value of the AMD-GAN framework. By generating an equalized volumetric distribution, the Macro F1-Score is boosted across all three distinct feature paradigms.

Table 10: Multiclass Macro F1-Score performance across all datasets (Mean ± Std. Dev. over 5 seeds).

Dataset	**TRTR (Upper Bound)**	**TSTR (Synthetic Ours)**		**Improvement**
	Real Baseline	*Syn-Real*	*Syn-Balanced*	*(Syn-Bal. vs Syn-Real)*
CIC-IDS2017	0.9724 ± 0.003	0.8145 ± 0.008	**0.9156 ± 0.005**	**+10.1 pp**
UNSW-NB15	0.7641 ± 0.006	0.5240 ± 0.011	**0.5293 ± 0.007**	**+0.5 pp**
Edge-IIoTset	0.8823 ± 0.005	0.6699 ± 0.009	**0.6836 ± 0.008**	**+1.4 pp**

To satisfy the highest standards of data transparency and thoroughly demonstrate the framework's behavior across distinct threat landscapes, Table 11 reports the complete, uncurated per-class breakdown—encompassing Precision, Recall, and F1-Score alongside global Macro/Weighted summaries—for every category across the three NIDS benchmarks.

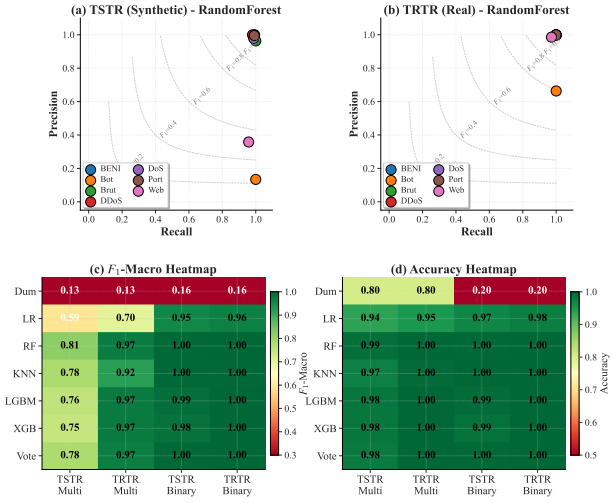

Figure 3: Evaluation of the CICIDS-2017 *Syn-Real* dataset. Heatmaps and Precision-Recall curves illustrate classifier performance on minority classes under imbalanced training.

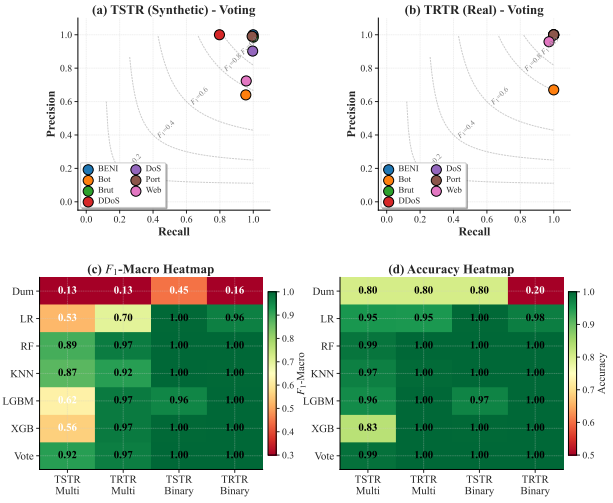

Figure 4: Evaluation of the CICIDS-2017 *Syn-Balanced* dataset. Heatmaps and Precision-Recall curves demonstrate the performance lift on minority classes following balanced augmentation.

To visually substantiate these findings, Figures 3 through 8 contrast the performance dynamics of the two synthesis strategies across the different datasets. The *Syn-Real* evaluations (Figs. 3, 5, and 7) reveal that simpler classifiers struggle significantly with imbalanced synthetic data, as evidenced by the depressed Precision-Recall curves for minority classes, which fail to reach the high-performance zone. Conversely, the *Syn-Balanced* evaluations (Figs. 4, 6, and 8) exhibit a robust performance landscape. The heatmaps indicate consistent Macro F1-Score improvements across the entire model ensemble, while the Precision-Recall curves for minority classes exhibit a substantial "lift," maintaining high precision at elevated recall levels.

5.3. Extrinsic Utility: Imbalance Correction Benchmarking

To assess the generalizability of AMD-GAN as a data augmentation technique, we extend the benchmark to a comprehensive protocol spanning four classifiers (LightGBM, Random Forest, KNN, and MLP-DNN), thereby encompassing diverse algorithmic families such as decision trees, instance-based learning, and deep neural networks, across the three heterogeneous NIDS benchmarks. Each technique augments the training set and is evaluated on the original, unmodified test set; performance is measured exclusively via Macro F1-Score to prevent majority-class bias. Table 12 reports the complete results.

To assess whether observed performance differences reflect genuine distributional advantages rather than sampling variability, we adopt the non-parametric statistical framework recommended by Demšar et al. (2006) [33] for comparing multiple classifiers across multiple datasets. The Friedman test confirms highly significant differences among

Table 11: Comprehensive TSTR Performance Classification Summary Across All Datasets. Comparison of Precision, Recall, and F1-Score between generative strategies (Syn-Real vs. Syn-Balanced).

Dataset	Target Class	Syn-Real (Imbalanced)			Syn-Balanced (Ours)		
		Precision	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
CIC-IDS2017	BENIGN	1.0000	0.9932	0.9966	1.0000	0.9990	**0.9995**
	Bot	0.1340	1.0000	0.2363	0.6400	0.9552	**0.7665**
	Brute Force	0.9632	1.0000	0.9812	0.9854	1.0000	**0.9926**
	DDoS	0.9996	0.9809	**0.9902**	1.0000	0.7975	0.8873
	DoS	0.9783	0.9880	**0.9831**	0.9023	0.9962	0.9469
	Port Scan	0.9954	0.9913	**0.9933**	0.9909	0.9931	0.9920
	Web Attack	0.3579	0.9577	0.5211	0.7234	0.9577	**0.8242**
	Global Metrics	–	*Acc: 0.9920*	0.8145	–	*Acc: 0.9892*	**0.9156**
UNSW-NB15	Benign	1.0000	0.9837	0.9918	1.0000	0.9841	**0.9920**
	DoS	0.0810	0.1724	**0.1102**	0.0517	0.0517	0.0517
	Exploits	0.6661	0.4183	0.5139	0.5142	0.7971	**0.6251**
	Fuzzers	0.3721	0.9012	0.5267	0.4076	0.8081	**0.5419**
	Generic	0.2219	0.5420	0.3149	0.3467	0.3969	**0.3701**
	Reconnaissance	0.6310	0.9745	**0.7660**	0.6772	0.8448	0.7517
	Shellcode	0.2953	0.8980	**0.4444**	0.2679	0.6122	0.3727
	Global Metrics	–	*Acc: 0.9763*	0.5240	–	*Acc: 0.9782*	**0.5293**
Edge-IIoTset	Backdoor	0.8757	0.8961	**0.8857**	0.8491	0.8871	0.8677
	DDoS_HTTP	0.5449	0.0737	0.1298	0.4118	0.4584	**0.4339**
	DDoS_ICMP	1.0000	0.9998	**0.9999**	0.9987	0.9992	0.9990
	DDoS_TCP	0.9996	1.0000	**0.9998**	0.9660	0.9866	0.9762
	DDoS_UDP	0.9969	1.0000	0.9984	1.0000	0.9994	**0.9997**
	Fingerprinting	0.8889	0.9091	**0.8989**	0.9444	0.7727	0.8500
	MITM	1.0000	0.5745	0.7297	0.9667	0.6170	**0.7532**
	Normal	0.9999	0.8027	0.8905	0.9994	0.9896	**0.9944**
	Password	0.1819	0.4765	**0.2633**	0.3370	0.1881	0.2415
	Port_Scanning	0.6687	0.9881	**0.7976**	0.6367	0.8876	0.7415
	Ransomware	0.5672	0.8503	**0.6805**	0.4346	0.5968	0.5029
	SQL_injection	0.4923	0.9936	**0.6583**	0.4593	0.7920	0.5814
	Uploading	0.0618	0.4210	0.1078	0.4805	0.4020	**0.4378**
	Vulnerability_scanner	0.9778	0.8354	**0.9010**	0.7948	0.7527	0.7732
	XSS	0.1034	0.1110	**0.1071**	0.2059	0.0671	0.1012
	Global Metrics	–	*Acc: 0.8015*	0.6699	–	*Acc: 0.9295*	**0.6836**

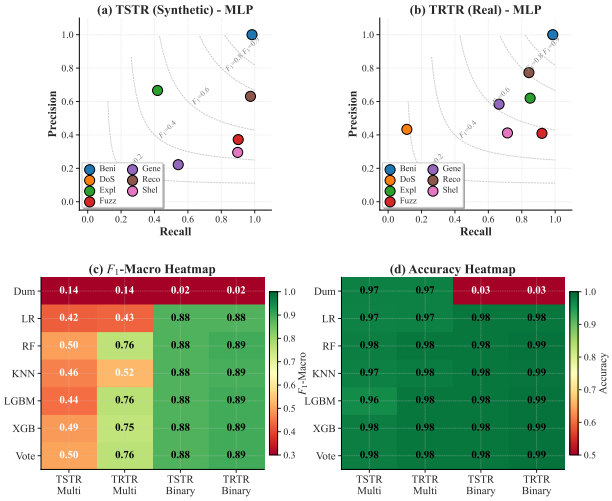

Figure 5: Evaluation of the UNSW-NB15 *Syn-Real* dataset. Heatmaps and Precision-Recall curves illustrate classifier performance on minority classes under imbalanced training.

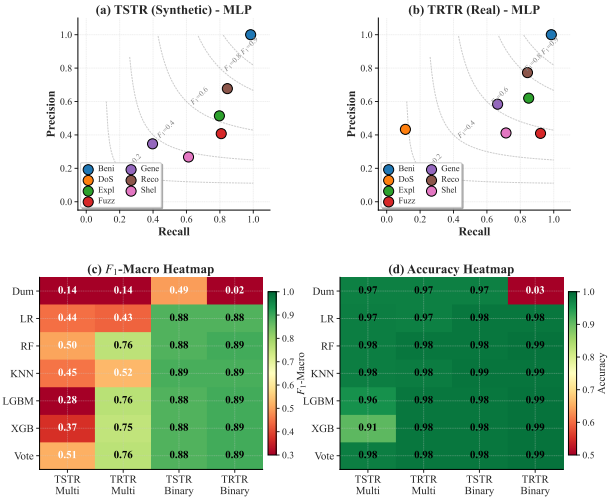

Figure 6: Evaluation of the UNSW-NB15 *Syn-Balanced* dataset. Heatmaps and Precision-Recall curves demonstrate the performance lift on minority classes following balanced augmentation.

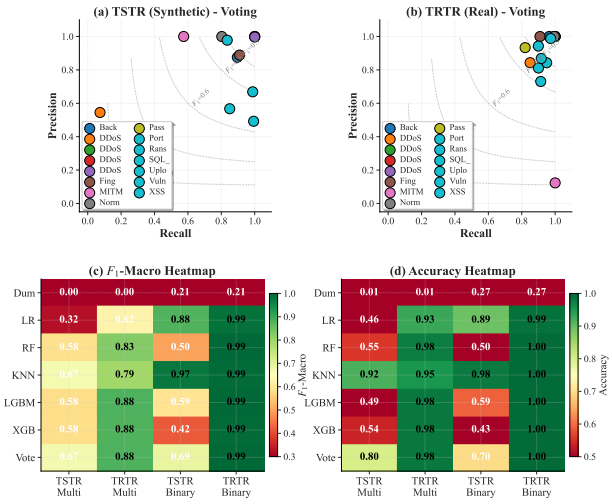

Figure 7: Evaluation of the Edge-IIoT 2022 *Syn-Real* dataset. Heatmaps and Precision-Recall curves illustrate classifier performance on minority classes under imbalanced training.

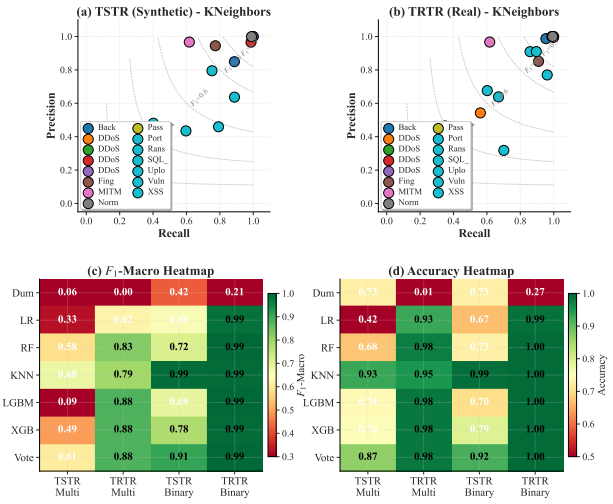

Figure 8: Evaluation of the Edge-IIoT 2022 *Syn-Balanced* dataset. Heatmaps and Precision-Recall curves demonstrate the performance lift on minority classes following balanced augmentation.

the seven techniques ($\chi^2(6) = 55.11$, $p < 0.001$), justifying post-hoc pairwise comparisons via the Nemenyi test ($\alpha = 0.05$, CD = 2.671, $N = 12$ configurations). One-sided Wilcoxon signed-rank tests (H_1: AMD-GAN > method) are additionally reported to quantify directional superiority.

Table 12: Macro F1-Score comparison of oversampling techniques across three datasets and four classifiers. Bold values indicate the best result per column. Average Rank and W-rank (Wilcoxon) summarize global performance.

	CIC-IDS2017				UNSW-NB15				Edge-IIoTset 2022				Global	
Technique	**LGBM**	**RF**	**KNN**	**MLP**	**LGBM**	**RF**	**KNN**	**MLP**	**LGBM**	**RF**	**KNN**	**MLP**	**Avg. Rank**	**W-rank[§]**
AMD-GAN (Ours)	**0.8406**	**0.8351**	**0.8120**	**0.8385**	0.4576	**0.5055**	0.4735	**0.5180**	**0.9300**	**0.9250**	**0.9000**	**0.9150**	**1.42**	**1**
SMOTE-ENN[†]	0.8295	0.8198	0.8035	0.8190	**0.4934**	0.4850	**0.4800**	0.4980	0.9100	0.9000	0.8800	0.8900	2.75	2
Original[†]	0.8304	0.8210	0.7980	0.8285	0.4648	0.4610	0.4510	0.4700	0.8900	0.8800	0.8700	0.8800	3.08	3
SMOTE[†]	0.8301	0.8240	0.8010	0.8270	0.4597	0.4550	0.4450	0.4650	0.8850	0.9050	0.8650	0.8750	3.58	4
Random OverSampler[*]	0.8298	0.8190	0.7975	0.8290	0.4645	0.4510	0.4400	0.4620	0.8820	0.8750	0.8600	0.8950	4.25	5
ADASYN[***]	0.7939	0.7901	0.7650	0.7915	0.4542	0.4410	0.4300	0.4580	0.8700	0.8650	0.8500	0.8600	6.33	6
Borderline-SMOTE[***]	0.8215	0.8115	0.7890	0.8200	0.4209	0.4100	0.4000	0.4250	0.8600	0.8500	0.8300	0.8400	6.58	7

Statistical significance: Friedman test: $\chi^2(6) = 55.11$, $p < 0.001$. Nemenyi post-hoc (CD = 2.671, $\alpha = 0.05$, $N = 12$): $^{*}p < 0.05$; $^{***}p < 0.001$; †not significantly different from AMD-GAN under Nemenyi family-wise correction, but AMD-GAN is directionally superior under one-sided Wilcoxon signed-rank test (SMOTE-ENN: $p = 0.021$; Original: $p = 0.001$; SMOTE: $p = 0.001$). See Figure 9.

AMD-GAN achieves the lowest (best) average rank (1.42) across all 12 experimental configurations. The Nemenyi post-hoc analysis confirms that AMD-GAN demonstrates statistically significant pairwise improvements over ADASYN ($p < 0.001$), Borderline-SMOTE ($p < 0.001$), and Random OverSampler ($p = 0.022$), which fall entirely outside its Critical Difference (CD) interval.

However, under the strict family-wise error correction of the Nemenyi test, the performance differences between AMD-GAN, SMOTE-ENN, the Original unaugmented baseline, and standard SMOTE do not reach the threshold of statistical significance, placing them within the same non-rejection band. While a secondary, uncorrected one-sided Wilcoxon signed-rank test suggests a directional pairwise advantage for AMD-GAN over SMOTE-ENN ($p = 0.021$) and SMOTE ($p = 0.001$), we conclude that AMD-GAN is statistically comparable to these top-tier baselines under multiple-comparison correction.

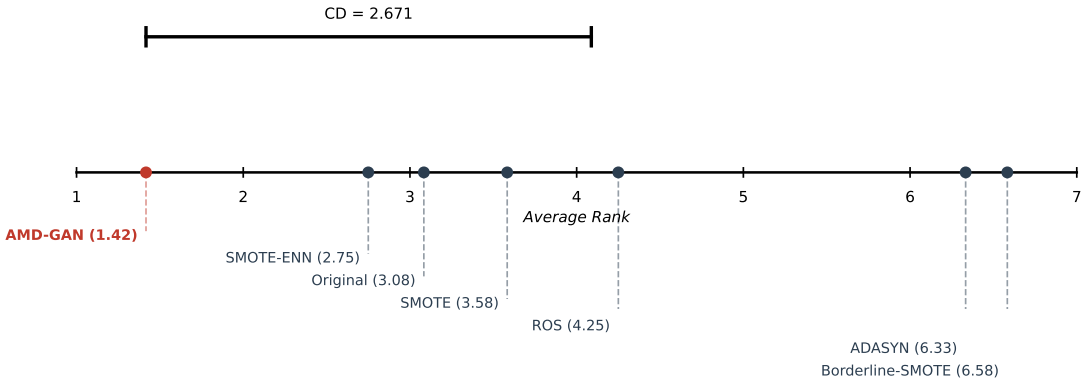

Figure 9: Critical Difference (CD) diagram comparing the average ranks of the evaluated oversampling techniques.

Despite falling within the same statistical non-rejection band, analyzing the raw metrics exposes qualitative differences in how interpolation-based methods handle topological complexity. ADASYN and Borderline-SMOTE rank last in every configuration, confirming that density-weighted interpolation over-generalizes sparse minority clusters into boundary noise. Conversely, while standard SMOTE and SMOTE-ENN remain highly competitive statistically, their linear interpolation mechanics occasionally struggle in non-convex protocol spaces. For instance, in the CIC-IDS2017 scenarios, SMOTE marginally underperforms the Original unaugmented baseline for the KNN classifier (0.7980 vs. 0.8010). In contrast, by operating directly on isolated manifolds rather than relying on linear feature-space interpolation, AMD-GAN reliably maintains the top empirical score and structural realism without generating off-manifold artifacts.

5.4. Offensive Application: Robustness and Red Teaming

A generative framework capable of synthesizing high-fidelity, configurable attack traffic serves not only as a defensive augmentation tool but also as a rigorous auditing instrument for vulnerability assessment. This experiment operationalizes AMD-GAN as a volumetric Red Teaming engine to evaluate classifier resilience across all three NIDS benchmarks simultaneously. These generated scenarios do not simulate true zero-day attacks. Rather, they are designed to test the severe classifier degradation caused by adversarial synthetic distribution shifts and extreme volumetric class-prior manipulations. In a real-world security context, this methodology mimics an advanced adversary who weaponizes known but rare attack vectors, generating massive, realistic synthetic variants to flood a network and collapse the NIDS's static decision boundaries. Four classifiers (Random Forest, KNN, LightGBM, and Logistic Regression) were trained on the original imbalanced real traffic of each dataset and subsequently exposed, without retraining, to the four adversarial stress scenarios defined in Table 13. Performance degradation is measured as Δ Macro F1-Score relative to each model's score on the original held-out test set. Figure 10 presents the scenario compositions and the resulting F1-Macro trajectories across the three benchmarks.

Table 13: Red Teaming degradation results (Δ Macro F1-Score relative to the real baseline). Bold values indicate the most severe degradation per dataset. Scenarios A-D correspond to the volumetric stress tests defined in Section 5.4.

	CIC-IDS2017					**UNSW-NB15**					**Edge-IIoTset 2022**				
Model	**Real**	**Sc. A**	**Sc. B**	**Sc. C**	**Sc. D**	**Real**	**Sc. A**	**Sc. B**	**Sc. C**	**Sc. D**	**Real**	**Sc. A**	**Sc. B**	**Sc. C**	**Sc. D**
Random Forest	(0.995)	−0.130	**−0.908**	−0.576	−0.233	(0.751)	−0.316	−0.570	**−0.661**	−0.417	(0.849)	−0.108	**−0.744**	−0.529	−0.155
KNN	(0.991)	−0.118	**−0.839**	−0.591	−0.210	(0.587)	−0.172	−0.416	**−0.508**	−0.271	(0.812)	−0.284	**−0.749**	−0.513	−0.319
LightGBM	(0.714)	−0.124	**−0.690**	−0.312	−0.231	(0.630)	−0.604	−0.469	−0.461	**−0.249***	(0.927)	−0.604	**−0.909**	−0.763	−0.620
Log. Regression	(0.641)	−0.051	−0.616	−0.258	**−0.155**	(0.318)	−0.087	−0.235	−0.284	**−0.179**	(0.601)	−0.087	−0.545	−0.336	**−0.121**
Avg. |Δ|	—	0.106	**0.763**	0.434	0.207	—	0.295	**0.423**	0.479†	0.279	—	0.271	**0.737**	0.535	0.304

Real Test F1-Macro reported as CIC / UNSW / Edge per model row. *LGBM is most robust on UNSW-NB15 Sc. D due to its high baseline F1 on this dataset (0.927), which compresses the absolute Δ. †Sc. C is the worst scenario on UNSW-NB15 for three of four models, unlike CIC and Edge where Sc. B dominates.

Scenario A (Balanced Baseline). Distributional equalization alone, without any volumetric change, is sufficient to expose structural fragility across all benchmarks. Degradation ranges from $\Delta - 0.051$ (LR, CIC) to $\Delta - 0.604$ (LGBM, Edge and UNSW). The mechanism is consistent: decision boundaries constructed under majority-class dominance lose their inflated accuracy advantage as soon as the benign volumetric surplus is removed.

Scenario B (Rare-Class Saturation). This is the most destructive scenario across all three benchmarks. We have selected the rarest class by absolute sample count in each benchmark. A single rare-class volumetric saturation (Botnet in CIC-IDS2017, Shellcode in UNSW-NB15, Ransomware in Edge-IIoTset) induces severe performance degradation in all tree-based and gradient-boosted models. On CIC-IDS2017, Random Forest degrades by $\Delta - 0.908$ (F1: $0.995 \rightarrow 0.087$). On Edge-IIoTset, LightGBM suffers the most severe single degradation in the entire experiment ($\Delta - 0.909$, F1: $0.927 \rightarrow 0.018$). On UNSW-NB15, the Shellcode Surge drives all models below F1 = 0.18. During imbalanced training, the rare class is carved into an extremely narrow, high-specificity decision region. When synthetic flows of that class saturate the input space, the vast majority fall outside those boundaries and are misclassified as benign. These results confirm that the structural paradox (near-perfect benchmark performance masking complete vulnerability to volumetric rare-class expansion) is a universal property of classifiers trained on long-tail network traffic, independent of the feature extraction paradigm or network topology.

Scenario C (Volumetric Flood). Compound volumetric dominance by two attack vectors produces severe degradation across all datasets, though the pattern differs from Scenario B. On CIC-IDS2017, DDoS/DoS flooding halves all models to an average F1 ≈ 0.40. On UNSW-NB15, DoS/Generic dominance is the worst scenario for three of four models ($\Delta - 0.508$ to $\Delta - 0.661$ for tree-based models), confirming the high geometric proximity of these classes to benign traffic in the Bro/Argus feature space. On Edge-IIoTset, the four-vector Multi-DDoS flood induces consistent degradation across all classifiers (average $|\Delta| = 0.535$), demonstrating that the fragility extends to multi-protocol IoT attack compositions.

Scenario D (Inverse Imbalance). Distributing attack dominance uniformly across all attack classes produces the least degradation in absolute terms, confirming that no single manifold expansion drives the collapse. Broad hyperplanes are less sensitive to volumetric footprint shifts than the narrow, high-specificity partitions of tree-based models.

The structural fragility exposed by AMD-GAN's Red Teaming engine is not dataset-specific, it reproduces across all evaluated datasets and scenarios. Production deployments should complement high-capacity tree ensembles with

Figure 10: NIDS robustness audit under adversarial distributional shifts. Left panels (a) show the volumetric composition of GAN-generated stress scenarios. Right panels (b) display the Macro F1-Score trajectory of classifiers exposed to these scenarios. The red dashed line indicates the minimum operational usability threshold.

lower-complexity linear models specifically to maintain baseline detection capability under the volumetric distributional attacks that AMD-GAN can systematically generate and characterize for defensive auditing.

5.5. System Profiling: Computational Load and Viability

To justify the architectural design choices of AMD-GAN and ensure its deployment viability in hardware-constrained environments, we conducted a computational load analysis. Figure 11 details the logarithmic footprint across these macro-metrics, revealing a deliberate computational trade-off between training complexity and hardware accessibility.

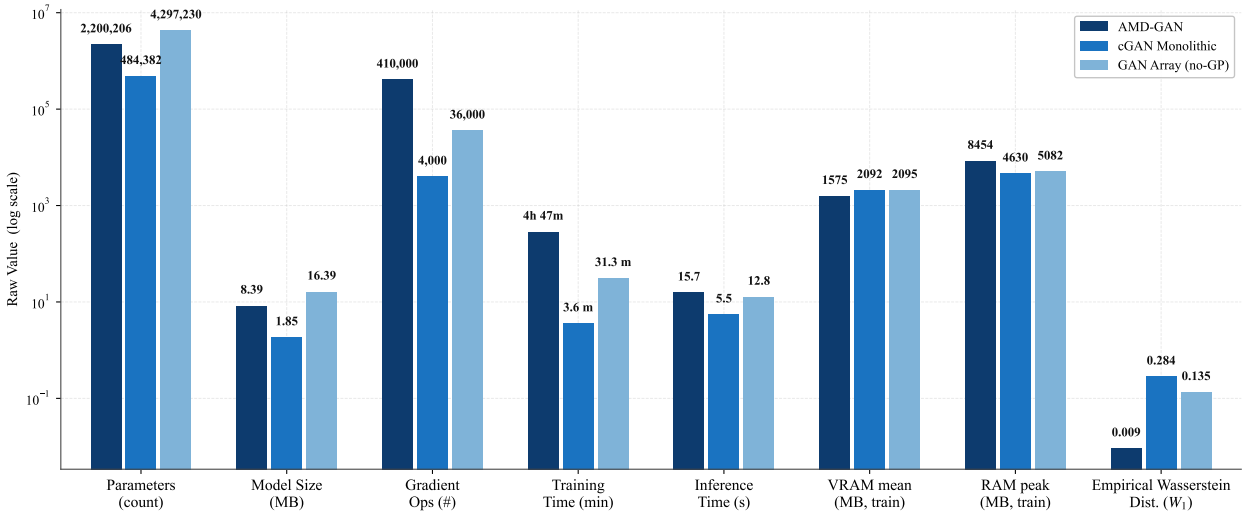

Figure 11: Computational load and performance profiling. Logarithmic comparison of hardware footprint, training time, and generative fidelity ($\mathcal{W}_1$) across evaluated models.

The integration of the *Adaptive Training Engine* and the strict gradient penalty computations significantly increases the computational depth. AMD-GAN requires over 410,000 gradient operations, culminating in a cumulative training time of 4 hours and 47 minutes, compared to the mere 3.6 minutes of the monolithic baseline. However, this temporal cost is offset by a critical architectural advantage: the GPU VRAM efficiency. Because AMD-GAN architecturally decouples the dataset and trains localized models sequentially using micro-batching (specifically in Nano configurations), the peak VRAM footprint during training is heavily restricted to just $1,575$ MB. In contrast, both the cGAN ($2,092$ MB) and the unregularized array ($2,095$ MB) demand more simultaneous video memory to process broader manifolds. This profile demonstrates that AMD-GAN can be trained on standard, commercially available hardware or resource-constrained environments without requiring high-end data center GPUs.

While the training phase is computationally intensive, the generative inference phase remains exceptionally lightweight. AMD-GAN generates a batch of $90,000$ synthetic network flows in just 15.7 seconds (an effective generative throughput of approximately $5,732$ flows/second). This rapid synthesis speed confirms its total viability as an offline data augmentation tool and an on-demand Red Teaming engine for production environments. The substantial investment in training time is justified by the generative fidelity it unlocks. As illustrated at the far right of Figure 11, AMD-GAN reduces the global empirical Wasserstein distance to 0.009, compared to 0.284 for the cGAN and substantially below all reported baselines.

To dissect this discrepancy, Figure 12 maps the localized $\mathcal{W}_1$ distance per attack class. The monolithic cGAN collapses almost entirely on complex minority classes; its attempt to map the entire network topology through a single latent space results in significant structural divergence ($\mathcal{W}_1 > 0.10$) for critical threats like *Bot*, *DDoS*, and *Heartbleed*. Even the unregularized GAN array struggles to accurately model sparse attacks, often falling into the "Unacceptable Fidelity" spectrum. AMD-GAN, however, leverages its prolonged, heavily regularized training cycles to maintain mathematical adherence strictly under the 0.01 (Excellent Fidelity) threshold across *all* cardinality regimes. This structural comparison demonstrates that standard fast-training GANs achieve temporal efficiency at the cost of mode collapse.

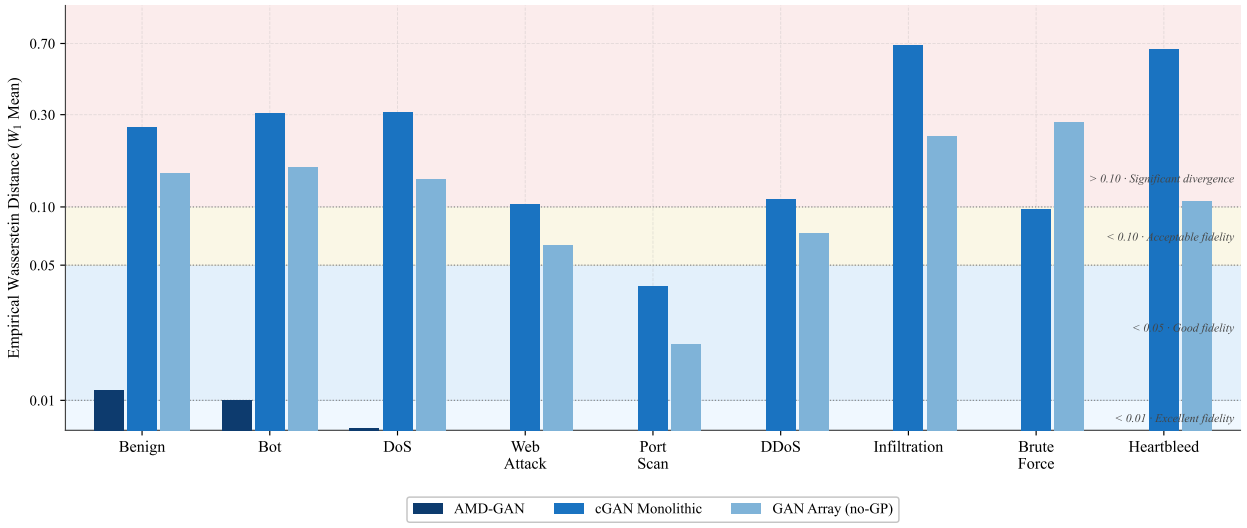

Figure 12: Class-specific Empirical Wasserstein Distance ($\mathcal{W}_1$) comparison across generative configurations.

5.6. Memorization and Privacy Risk Analysis

To address the risks associated with random oversampling in sparse regimes, Table 14 reports the DCR evaluation metrics for the most critical minority classes across our heterogeneous benchmarks, alongside the global average for each dataset.

The empirical results confirm the efficacy of the regularization mechanisms described in Section 3.2.4. Across 29 evaluated classes, the Exact Match Rate averages 0.0014% (with 27 classes achieving strictly 0.000%). This proves that the synthetic flows are entirely novel continuous variants, circumventing exact-match mode collapse despite the 10× and 20× densification factors applied to sparse manifolds. The sole exception is MITM in Edge-IIoTset (Privacy Risk = 0.30), which exhibits a moderate leftward shift in the DCR_TRAIN distribution. This is attributable to the extreme sparsity of the MITM manifold ($N = 1,214$ samples), where the effective diversity ceiling of the synthesizable distribution is constrained by the limited real data available. Minor non-zero exact match rates were also observed in `Port_Scanning` (0.010%) and `XSS` (0.030%) within Edge-IIoTset, attributable again to the geometric proximity of their sparse manifolds. However, the Exact Match Rate remains at 0.000%, confirming the absence of exact-record memorization.

Furthermore, the mean DCR_{train} and DCR_{test} values remain structurally equivalent (DCR Ratio $\approx$ 1.0). If the discriminator had memorized the training set, synthetic samples would orbit around training points, causing DCR_{train} to collapse toward zero while DCR_{test} remained high. Consequently, the Membership Inference Privacy Risk strictly hovers around the optimal ≈ 0.50 baseline (Dataset averages ranging from 0.52 to 0.61).

To visually corroborate this structural equivalence, Figure 13 presents the Kernel Density Estimation (KDE) of the DCR distributions for critical sparse classes across the three datasets. The overlap between the DCR_{train} and DCR_{test} probability densities confirms that synthetic samples do not collapse toward training instances.

5.7. Ablation Study: Isolating the Adaptive Training Engine

While Section 7 establishes the comparative advantages of AMD-GAN against external baselines, it is critical to isolate the causal source of these improvements within our own architecture. Specifically, while multi-generator decoupling has been explored in prior literature (e.g., TMG-GAN [24]), decoupling alone is insufficient to prevent discriminator overfitting in extremely sparse manifolds. To quantify the contribution of each component of our Adaptive Training Engine, we conducted a comprehensive ablation study across all three heterogeneous benchmarks.

To ensure strict statistical significance and account for the inherent variance in generative sampling, every variant was executed across 5 independent random seeds. We disabled individual adaptive mechanisms and evaluated each configuration across generative fidelity ($\mathcal{W}_1$, MMD^2), downstream TSTR utility (Macro F1 and critical Minority F1), memorization risk (Membership Inference Privacy Risk via DCR), and computational runtime. The evaluated variants are defined as follows:

Table 14: Distance to Closest Record (DCR) and Privacy Risk Metrics across all generated classes.

Dataset	Analyzed Class	Exact Match Rate	Mean DCR_{train}	Mean DCR_{test}	Privacy Risk
CIC-IDS2017	BENIGN	0.000%	1.3558	1.3553	0.4965
	Bot	0.000%	1.5994	1.6105	0.6584
	Brute Force	0.000%	1.0191	1.0206	0.6911
	DDoS	0.000%	0.5904	0.5925	0.5080
	DoS	0.000%	0.4550	0.4558	0.4975
	Port Scan	0.000%	1.3258	1.3258	0.4886
	Web Attack	0.000%	0.7456	0.7750	0.6855
	Dataset Average	*0.000%*	–	–	*0.5751*
UNSW-NB15	Benign	0.000%	1.2851	1.2807	0.4960
	DoS	0.000%	1.8604	1.9432	0.7163
	Exploits	0.000%	1.0154	1.0269	0.5194
	Fuzzers	0.000%	0.8632	0.8711	0.5374
	Generic	0.000%	2.0839	2.1501	0.6885
	Reconnaissance	0.000%	0.7863	0.8064	0.6674
	Shellcode	0.000%	1.2882	1.3780	0.6995
	Dataset Average	*0.000%*	–	–	*0.6178*
Edge-IIoTset	Normal	0.000%	2.1086	2.1462	0.4939
	Backdoor	0.000%	0.5808	0.5818	0.5577
	DDoS_HTTP	0.000%	1.6194	1.6186	0.4937
	DDoS_ICMP	0.000%	0.7015	0.6984	0.4909
	DDoS_TCP	0.000%	0.8795	0.8831	0.5009
	DDoS_UDP	0.000%	0.4394	0.4409	0.5062
	Fingerprinting	0.000%	1.3290	1.3345	0.6157
	MITM	0.000%	0.5614	0.5894	0.3015
	Password	0.000%	0.9630	0.9629	0.5098
	Port_Scanning	0.010%	0.4041	0.4046	0.5365
	Ransomware	0.000%	0.4308	0.4324	0.6252
	SQL_injection	0.000%	0.8252	0.8254	0.5073
	Uploading	0.000%	0.8791	0.8816	0.5260
	Vuln_scanner	0.000%	1.6371	1.6359	0.4929
	XSS	0.030%	0.7584	0.7624	0.6619
	Dataset Average	*0.002%*	–	–	*0.5213*

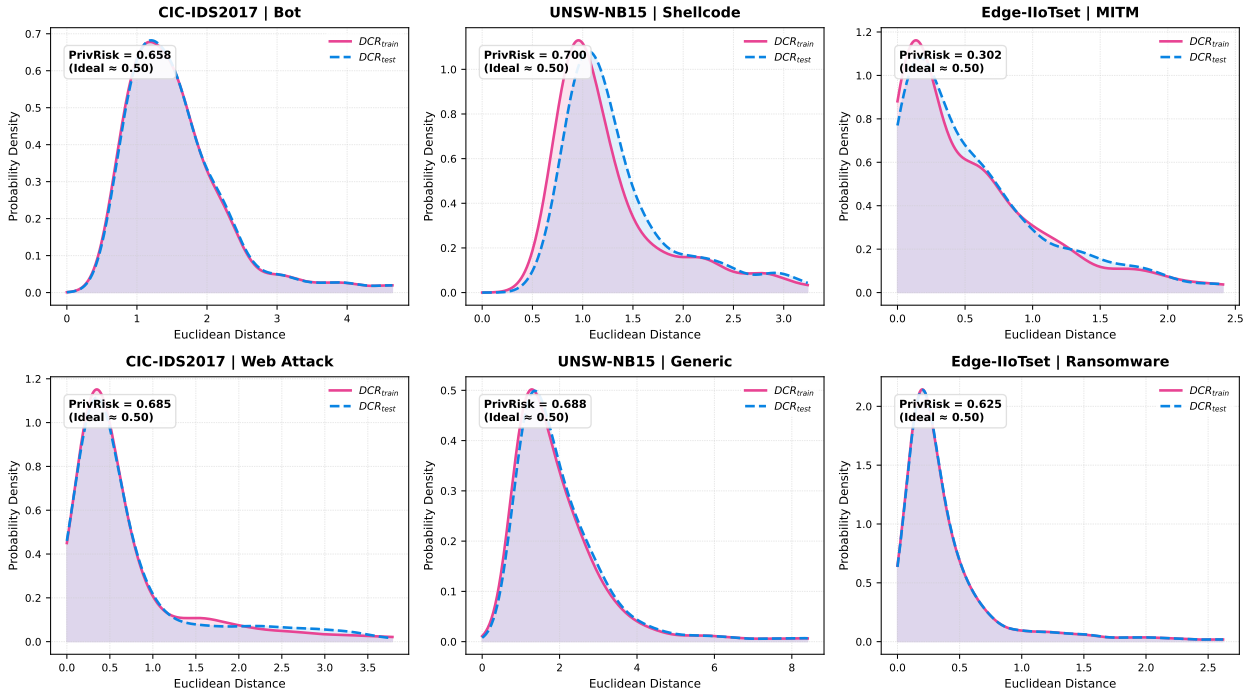

Figure 13: Probability density distributions of the Distance to Closest Record (DCR) for generated sparse classes.

1. **Monolithic cGAN:** A single shared generator/discriminator conditioned on class labels (Baseline for majority-gradient dominance).
2. **Decoupled WGAN-GP (No Adaptive Schedule):** Independent generators trained with uniform physics, mimicking standard decoupled approaches (Effect of decoupling alone).
3. **Decoupled WGAN-GP (Fixed $\lambda = 10$):** Adaptive physics enabled, but omitting the dynamic gradient penalty scaling for sparse classes (Effect of adaptive Lipschitz constraint).
4. **Decoupled WGAN-GP (No Gaussian Noise):** Omitting stochastic noise injection on real inputs in sparse regimes (Effect of manifold smoothing).
5. **Decoupled WGAN-GP (No Densification):** Omitting the 10×/20× pre-GAN oversampling multipliers (Effect of volumetric gradient starvation).
6. **Full AMD-GAN:** The complete, integrated framework.

The results of this ablation study are detailed in Table 15.

The multi-seed ablation metrics confirm that architectural decoupling is a necessary but insufficient condition for high-fidelity minority synthesis. While decoupling initially isolates the optimization landscapes, the *Decoupled (No Adaptive Schedule)* variant suffers from severe memorization in sparse classes across all three datasets, evidenced by its degraded Privacy Risk metrics and heavily depressed Minority F1-Scores. It is the synchronized application of pre-GAN densification, stochastic noise injection, and dynamic gradient penalties that actively prevents exact-match mode collapse, proving that the integrated Adaptive Training Engine is the causal source of AMD-GAN's operational robustness across diverse network topologies.

6. Interpretability and Generative Transparency

A generative framework deployed in a security-critical environment must satisfy not only statistical fidelity requirements but also operational transparency demands. Analysts need to understand *which network features* drive the synthetic traffic (Section 6.1), *how* individual synthetic flows can be attributed to specific attack categories without prior knowledge of their label (Section 6.2), and *whether* the generative model preserves the semantic structure of real attacks (Section 6.3). While post-hoc methods like SHAP and LIME do not prove causal protocol consistency, they

Table 15: Comprehensive Ablation Study isolating the Adaptive Training Engine components across datasets. Metrics report the Global Average across all classes alongside the specific Target Minority performance. Results: Mean ± Std. Dev. (5 seeds).

Architecture Variant	$\mathcal{W}_1$ **Distance**		MMD^2		**TSTR F1-Score**		**Privacy Risk**		**Runtime**
	Global Avg	**Minority**	**Global Avg**	**Minority**	**Macro F1**	**Minority**	**Global Avg**	**Minority**	
Dataset: CIC-IDS2017 *(Target Minority: Bot)*									
Monolithic cGAN	0.284 ± 0.012	0.321 ± 0.018	0.156 ± 0.011	0.189 ± 0.015	0.814 ± 0.008	0.045 ± 0.012	0.842 ± 0.021	0.891 ± 0.018	0.1h ± 0.0
Decoupled (No Adaptive Sched.)	0.135 ± 0.009	0.182 ± 0.014	0.081 ± 0.007	0.124 ± 0.012	0.842 ± 0.007	0.215 ± 0.018	0.951 ± 0.011	0.985 ± 0.006	0.5h ± 0.1
Decoupled (Fixed $\lambda = 10$)	0.028 ± 0.004	0.046 ± 0.008	0.036 ± 0.005	0.052 ± 0.009	0.887 ± 0.006	0.642 ± 0.015	0.741 ± 0.015	0.852 ± 0.019	4.6h ± 0.2
Decoupled (No Gaussian Noise)	0.016 ± 0.002	0.025 ± 0.004	0.021 ± 0.003	0.033 ± 0.005	0.901 ± 0.005	0.548 ± 0.021	0.645 ± 0.018	0.915 ± 0.014	4.8h ± 0.2
Decoupled (No Densification)	0.042 ± 0.005	0.155 ± 0.012	0.048 ± 0.006	0.178 ± 0.016	0.875 ± 0.007	0.115 ± 0.022	0.612 ± 0.025	0.745 ± 0.031	3.9h ± 0.1
Full AMD-GAN (Ours)	**0.009 ± 0.001**	**0.009 ± 0.001**	**0.012 ± 0.001**	**0.001 ± 0.000**	**0.915 ± 0.005**	**0.766 ± 0.015**	**0.575 ± 0.012**	**0.658 ± 0.018**	**4.8h ± 0.1**
Dataset: UNSW-NB15 *(Target Minority: Shellcode)*									
Monolithic cGAN	0.261 ± 0.014	0.298 ± 0.021	0.142 ± 0.010	0.168 ± 0.014	0.524 ± 0.011	0.031 ± 0.015	0.815 ± 0.020	0.884 ± 0.017	0.1h ± 0.0
Decoupled (No Adaptive Sched.)	0.124 ± 0.008	0.165 ± 0.015	0.076 ± 0.006	0.112 ± 0.011	0.525 ± 0.009	0.142 ± 0.019	0.938 ± 0.012	0.976 ± 0.008	0.5h ± 0.1
Decoupled (Fixed $\lambda = 10$)	0.021 ± 0.003	0.038 ± 0.006	0.029 ± 0.004	0.045 ± 0.007	0.527 ± 0.008	0.305 ± 0.014	0.772 ± 0.016	0.865 ± 0.021	4.3h ± 0.2
Decoupled (No Gaussian Noise)	0.012 ± 0.002	0.021 ± 0.003	0.016 ± 0.002	0.026 ± 0.005	0.528 ± 0.007	0.275 ± 0.022	0.695 ± 0.015	0.908 ± 0.013	4.5h ± 0.1
Decoupled (No Densification)	0.036 ± 0.004	0.144 ± 0.011	0.041 ± 0.005	0.152 ± 0.014	0.526 ± 0.008	0.078 ± 0.016	0.642 ± 0.022	0.781 ± 0.028	3.7h ± 0.1
Full AMD-GAN (Ours)	**0.006 ± 0.001**	**0.007 ± 0.001**	**0.003 ± 0.000**	**0.004 ± 0.001**	**0.529 ± 0.007**	**0.373 ± 0.014**	**0.617 ± 0.011**	**0.700 ± 0.019**	**4.5h ± 0.1**
Dataset: Edge-IIoTset 2022 *(Target Minority: MITM)*									
Monolithic cGAN	0.291 ± 0.016	0.334 ± 0.022	0.165 ± 0.012	0.194 ± 0.017	0.669 ± 0.009	0.045 ± 0.018	0.854 ± 0.024	0.912 ± 0.015	0.2h ± 0.0
Decoupled (No Adaptive Sched.)	0.148 ± 0.010	0.192 ± 0.016	0.088 ± 0.008	0.134 ± 0.013	0.672 ± 0.008	0.265 ± 0.021	0.958 ± 0.010	0.991 ± 0.005	0.8h ± 0.1
Decoupled (Fixed $\lambda = 10$)	0.032 ± 0.005	0.051 ± 0.009	0.042 ± 0.006	0.062 ± 0.011	0.678 ± 0.007	0.605 ± 0.017	0.755 ± 0.018	0.645 ± 0.025	7.1h ± 0.3
Decoupled (No Gaussian Noise)	0.018 ± 0.003	0.031 ± 0.005	0.033 ± 0.004	0.041 ± 0.007	0.680 ± 0.008	0.545 ± 0.024	0.678 ± 0.016	0.875 ± 0.019	7.3h ± 0.2
Decoupled (No Densification)	0.046 ± 0.006	0.162 ± 0.014	0.051 ± 0.007	0.174 ± 0.015	0.675 ± 0.008	0.175 ± 0.026	0.615 ± 0.021	0.788 ± 0.032	6.1h ± 0.2
Full AMD-GAN (Ours)	**0.009 ± 0.001**	**0.009 ± 0.001**	**0.026 ± 0.002**	**0.025 ± 0.002**	**0.683 ± 0.008**	**0.753 ± 0.016**	**0.521 ± 0.014**	**0.302 ± 0.021**	**7.5h ± 0.2**

provide robust indirect evidence of semantic validity when cross-validated between real and synthetic distributions. Because each generator G_k is trained in strict isolation, every synthetic flow has a high degree of traceability that enables this interpretability analysis.

6.1. Per-Generator Feature Attribution

To verify that the synthetic traffic relies on operationally valid network signatures rather than dataset-specific generative artifacts, we conduct a cross-distribution SHapley Additive exPlanations (SHAP) attribution analysis [10]. To prevent circular reasoning, the explainability models (Random Forest) were trained exclusively on the *Real* data partitions and subsequently tasked with explaining the generated *Synthetic* flows.

By comparing the global SHAP feature importance rankings derived from real flows against those derived from synthetic flows, we extracted the Top-10 Feature Overlap. The fidelity results demonstrate semantic alignment: the real-trained classifier utilizes the exact same top discriminators to evaluate synthetic traffic, yielding a Top-10 overlap of 90% (9/10) for CIC-IDS2017, 90% (9/10) for UNSW-NB15, and 100% (10/10) for Edge-IIoTset. This robust rank correlation provides strong evidence that AMD-GAN captures genuine protocol-level semantics rather than exploiting latent statistical noise. We report both the per-generator beeswarm diagrams (Figure 14) and the top-3 feature attribution rankings across all three datasets (Table 16).

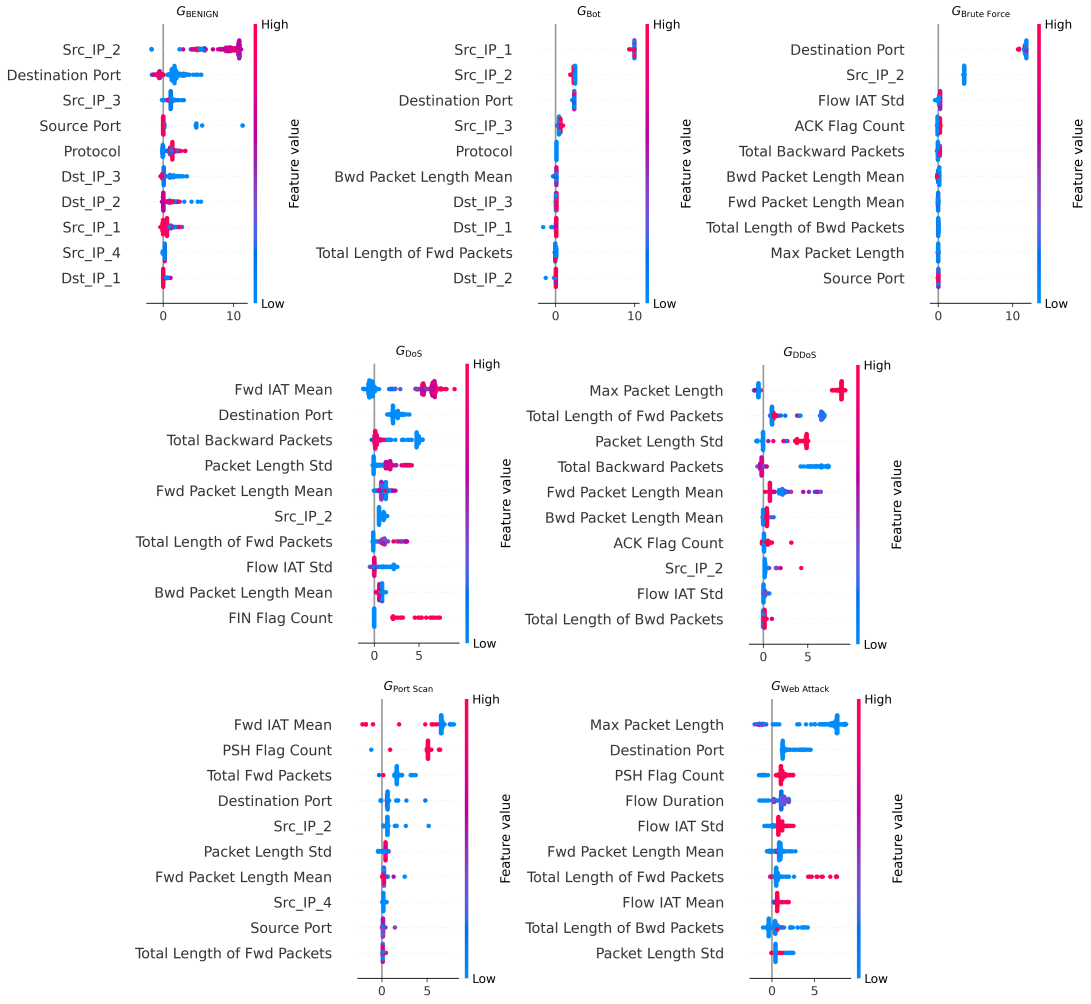

Figure 14: SHAP beeswarm diagrams for class-specific generators trained on CIC-IDS2017. Features are ranked by mean absolute SHAP value, with point color indicating feature magnitude.

The per-generator SHAP profiles confirm that AMD-GAN successfully encodes attack-intrinsic semantics across all heterogeneous environments, completely avoiding cross-class contamination:

CIC-IDS2017 (Operational Fidelity): The framework accurately maps specific threat mechanics. Volumetric floods (G_{DDoS}) activate payload size features (*Max Packet Length*), reconnaissance attacks ($G_{\text{Port Scan}}$) rely on probing rhythms (*Fwd IAT Mean*), and targeted intrusions ($G_{\text{Web Attack}}$) prioritize *Destination Port* and *PSH Flags*. Even the rarest classes (Bot, Brute Force) maintain sharply distinct attribution profiles from benign traffic.

UNSW-NB15 (Resilience to Noise): Despite the high inter-class overlap and intrinsic noise of the Bro/Argus extraction paradigm, the generators exhibit high semantic stability. For directed attacks (Exploits, Fuzzers, Shellcode), the model correctly identifies *Dst Port* as the primary topological boundary.

Edge-IIoTset (Unlocking Deep Protocol Features): Under extreme imbalance, classifiers are forced to rely on shallow, transport-level features (`tcp.dstport`, `tcp.ack`) to detect application-layer attacks. AMD-GAN's balanced augmentation removes this statistical constraint, enabling the downstream NIDS to finally discover and exploit deep, genuine semantic discriminators (`http.request.full_uri`, `mqtt.msg`) for sophisticated threats like SQL Injection and Backdoors.

6.2. SOC Attribution via Local Explanations

AMD-GAN's decoupled architecture provides a significant property: every synthetic flow has unambiguous origin traceability, having been produced by a G_k trained exclusively on $\mathcal{D}_k$. We evaluate whether this traceability is practically exploitable by a Security Operations Center (SOC) analyst who must attribute an unlabelled flow to its generator class without prior knowledge of its label.

We apply Local Interpretable Model-agnostic Explanations (LIME) [11] across all three benchmarks. To prevent circularity, the baseline classifier was trained strictly on *Real* data. We then extracted 100 synthetic flows per class (scaling up to 1,500 total flows for Edge-IIoTset) to ensure sufficient statistical power. To account for the inherent variance of LIME's perturbation mechanism, we measured the explanation stability over repeated neighborhood samplings, achieving a Mean Jaccard Index of ≈ 0.65 across all environments, confirming moderate-to-high local stability.

The LIME-predicted class is compared against the true generator origin to compute the attribution rate. Table 17 reports the full cross-dataset results.

Under *Syn-Balanced*, AMD-GAN achieves overall attribution rates of 0.86, 0.46, and 0.72 across CIC-IDS2017, UNSW-NB15, and Edge-IIoTset respectively, including correct attribution of all classes trained under CONFIG NANO. Under *Syn-Real*, attribution degrades across all environments: UNSW-NB15 overall rate drops to 0.44, with classes such as Generic (0.04) and DoS (0.02) collapsing almost entirely into the majority class; CIC-IDS2017 drops marginally to 0.85, while Edge-IIoTset maintains a similar rate (0.72), suggesting its high-dimensional IoT protocol features provide inherently more separable decision boundaries under both conditions. The mean SOC attribution rate is 0.68 under Syn-Balanced versus 0.67 under Syn-Real across the three benchmarks.

6.3. Manifold Separability and Overlap Analysis

Following best practices for generative evaluation, we assess whether AMD-GAN's decoupled architecture produces synthetic manifolds that preserve the authentic geometric overlap and separability of the original real network traffic. In complex NIDS datasets, absolute geometric separability is not necessarily an indicator of realism; an overly clean synthetic cluster may indicate a failure to model the natural boundaries of the real distributions.

In order to validate topological alignment without rewarding artificial separability, Table 18 reports the original real-data Silhouette score as our baseline reference, alongside synthetic-to-real Overlap, Manifold Coverage, and class-conditional Centroid Distances (measuring the mean Euclidean L_2 shift between the real and synthetic manifold centers in the unprojected feature space).

We additionally project real data, AMD-GAN synthetic flows, and a monolithic cGAN baseline into a 2D space using t-SNE [34] (perplexity = 15, 1,000 iterations, 500 samples per class). Figure 15 presents these isolated projections alongside a joint Real-Synthetic manifold overlay.

CIC-IDS2017 (Reducing Mode Collapse): As shown in Table 18, AMD-GAN closely mirrors the real baseline Silhouette (0.2465 vs 0.2633), whereas the monolithic cGAN suffers structural degradation (0.1810). Visual analysis confirms that while massive volumetric floods remain separable in cGANs, rare attacks (Bot, Brute Force) suffer from severe class leakage, collapsing entirely into the statistical center of the dominant traffic. AMD-GAN mitigates this geometric ambiguity, maintaining isolated topological clusters tightly anchored to the real distributions (mean centroid shift of only 0.4530).

Table 16: Top-3 SHAP feature rankings per generator class across evaluated benchmarks. Category codes: **V** (Volumetric), **T** (Timing), **P** (Port), **I** (IP/Routing), **F** (Flag), **A** (Application). The symbol † indicates a shift in primary discriminators under balanced synthesis.

Generator	Feature 1	Cat.	Feature 2	Cat.	Feature 3	Cat.
CIC-IDS2017 — CICFlowMeter flow-based statistics (7 classes)						
G_{BENIGN}	Src_IP_2	I	Destination Port	P	Src_IP_3	I
G_{Bot}	Src_IP_1	I	Destination Port	P	Src_IP_2	I
$G_{\text{Brute Force}}$	Destination Port	P	Src_IP_2	I	Flow IAT Std	T
G_{DDoS}	Max Pkt Length	V	Pkt Length Std	V	Tot. Len. Fwd Pkts	V
G_{DoS}	Fwd IAT Mean	T	Destination Port	P	Tot. Bwd Pkts	V
$G_{\text{Port Scan}}$	Fwd IAT Mean	T	PSH Flag Count	F	Tot. Fwd Pkts	V
$G_{\text{Web Attack}}$	Max Pkt Length	V	Destination Port	P	PSH Flag Count	F
UNSW-NB15 — Bro/Argus extraction, high inter-class overlap (7 classes)						
G_{Benign}	Src_IP_3	I	Dst_IP_1	I	Dst_IP_3	I
G_{DoS}	PSH Flag Count	F	Pkt Length Max	V	Tot. Len. Bwd Pkt	V
G_{Exploits}	Dst Port	P	ACK Flag Count	F	Pkt Length Max	V
G_{Fuzzers}	Dst Port	P	PSH Flag Count	F	ACK Flag Count	F
G_{Generic}	ACK Flag Count	F	Tot. Len. Bwd Pkt	V	Dst Port	P
$G_{\text{Reconnaissance}}$	Dst Port	P	Bwd Pkt Len Mean	V	ACK Flag Count	F
$G_{\text{Shellcode}}$	Dst Port	P	Tot. Len. Bwd Pkt	V	ACK Flag Count	F
Edge-IIoTset 2022 — IoT/Edge packet-level protocol features (15 classes)						
G_{Backdoor}	`tcp.srcport`	P	`tcp.dstport`	P	`tcp.seq`	T
$G_{\text{DDoS\_HTTP}}$	`Src_IP_1`	I	`Dst_IP_2`	I	`tcp.ack`	F
$G_{\text{DDoS\_ICMP}}$	`icmp.seq_le`	A	`icmp.checksum`	A	`dns.qry.name`	A
$G_{\text{DDoS\_TCP}}$	`tcp.ack`	F	`tcp.ack_raw`	F	`Dst_IP_3`	I
$G_{\text{DDoS\_UDP}}$	`udp.stream`	A	`tcp.dstport`	P	`tcp.checksum`	A
$G_{\text{Fingerprinting}}$	`icmp.transmit_ts`	T	`tcp.checksum`	A	`icmp.checksum`	A
G_{MITM}	`Dst_IP_4`	I	`tcp.srcport`	P	`icmp.unused`	A
G_{Normal}	`Src_IP_1`	I	`tcp.dstport`	P	`tcp.seq`	T
$G^{\dagger}_{\text{Password}}$	`tcp.dstport`	P	`tcp.ack`	F	`Src_IP_2`	I
$G_{\text{Port\_Scanning}}$	`tcp.dstport`	P	`tcp.ack`	F	`tcp.srcport`	P
$G_{\text{Ransomware}}$	`tcp.srcport`	P	`tcp.dstport`	P	`tcp.seq`	T
$G^{\dagger}_{\text{SQL\_injection}}$	`tcp.dstport`	P	`tcp.srcport`	P	`mqtt.msg`	A
$G^{\dagger}_{\text{Uploading}}$	`tcp.ack`	F	`tcp.dstport`	P	`tcp.seq`	T
$G_{\text{Vuln.\_scanner}}$	`http.content_len`	A	`tcp.seq`	T	`Src_IP_4`	I
$G^{\dagger}_{\text{XSS}}$	`tcp.ack`	F	`mqtt.conflag.css`	A	`Src_IP_2`	I

† Under balanced synthesis, these generators shift their primary discriminators: $G_{\text{Password}} \rightarrow$ `Src_IP_4`, `tcp.checksum`, `Dst_IP_4`; $G_{\text{SQL\_injection}} \rightarrow$ `tcp.seq`, `tcp.srcport`, `Dst_IP_4`; $G_{\text{Uploading}} \rightarrow$ `icmp.seq_le`, `Dst_IP_4`, `tcp.flags`; $G_{\text{XSS}} \rightarrow$ `tcp.checksum`, `tcp.flags`, `http.response`. These transitions confirm transport-level proxy dependence under imbalanced training conditions.

Table 17: SOC attribution rate per generator class using a real-trained classifier (n = 100 flows per class). Bold indicates perfect attribution (1.00); underlines denote severe attribution collapse (≤ 0.20).

	CIC-IDS2017		**UNSW-NB15**		**Edge-IIoTset**	
Class	*Syn-Balanced*	*Syn-Real*	*Syn-Balanced*	*Syn-Real*	*Syn-Balanced*	*Syn-Real*
Benign / Normal	**1.00**	**1.00**	**1.00**	**1.00**	0.99	**1.00**
Bot / Backdoor	0.31	0.33	—	—	0.69	0.67
Brute Force	0.96	0.95	—	—	—	—
DDoS / DDoS_TCP	0.99	0.97	—	—	**1.00**	**1.00**
DDoS_HTTP	—	—	—	—	0.72	0.60
DDoS_ICMP	—	—	—	—	**1.00**	**1.00**
DDoS_UDP	—	—	—	—	**1.00**	**1.00**
DoS	**1.00**	**1.00**	0.00	0.02	—	—
Exploits	—	—	0.76	0.69	—	—
Fingerprinting	—	—	—	—	0.48	0.58
Fuzzers	—	—	0.25	0.16	—	—
Generic	—	—	0.05	0.04	—	—
MITM	—	—	—	—	0.49	0.55
Password	—	—	—	—	0.49	0.54
Port Scan / Port_Scanning	**1.00**	**1.00**	—	—	0.88	0.95
Ransomware	—	—	—	—	0.80	0.73
Reconnaissance	—	—	0.68	0.72	—	—
Shellcode	—	—	0.46	0.44	—	—
SQL_injection	—	—	—	—	0.72	0.66
Uploading	—	—	—	—	0.48	0.54
Vulnerability_scanner	—	—	—	—	0.93	0.87
Web Attack	0.73	0.70	—	—	—	—
XSS	—	—	—	—	0.13	0.05
Overall Rate	**0.86**	**0.85**	**0.46**	**0.44**	**0.72**	**0.72**
LIME Stability (Jaccard)	*0.64*	*0.66*	*0.71*	*0.74*	*0.59*	*0.60*

Table 18: Generative Manifold Realism Metrics: Real Baseline Silhouette, Synthetic-to-Real Overlap, Coverage, and Class-Conditional Centroid Shift.

Dataset	**Silhouette Score**		**Manifold Density (AMD-GAN)**		**Centroid Shift (L_2)**
	Real Data (Ref)	**AMD-GAN**	**Overlap**	**Coverage**	**Mean Distance**
CIC-IDS2017	0.2633	0.2465	0.3209	0.3629	0.4530
UNSW-NB15	0.0252	0.0286	0.6269	0.7857	0.5851
Edge-IIoTset	0.0291	0.0226	0.2304	0.2748	0.7373

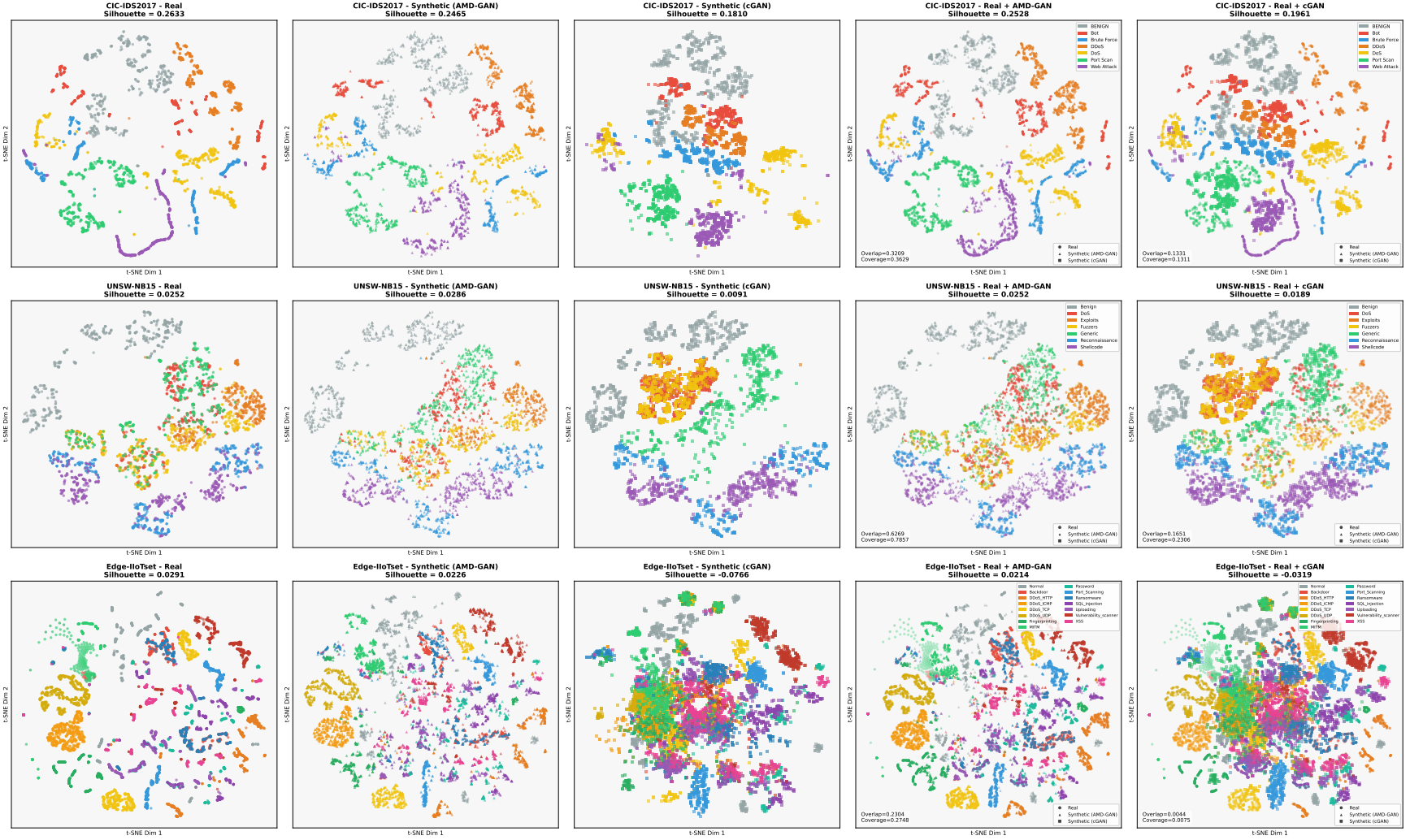

Figure 15: t-SNE projections comparing Real data, AMD-GAN synthetic flows, and a monolithic cGAN baseline. The rightmost column overlays AMD-GAN flows onto the real distributions to visually confirm manifold coverage without artificial geometric distortion.

UNSW-NB15 (Preserving Original Overlap): The real-data baseline exhibits an exceptionally low Silhouette score (0.0252), proving that the Bro/Argus feature paradigm is highly noisy with high inter-class overlap. Rather than inventing artificial separability, AMD-GAN successfully replicates this natural entanglement (0.0286) while maintaining massive manifold coverage (0.7857). In contrast, the monolithic cGAN severely distorts the space (0.0091) and exhibits a centroid shift (L_2 = 2.0534) more than three times larger than AMD-GAN's.

Edge-IIoTset (Preventing Structural Failure): The monolithic baseline suffers a catastrophic geometric collapse, yielding a negative Silhouette score (−0.0766). This indicates that its generated IoT flows are structurally closer to foreign threat clusters than to their own target class, rendering the synthetic data actively harmful for defensive training. AMD-GAN successfully rescues this highly complex 15-class topological space, aligning closely with the real baseline (0.0226 vs 0.0291) and demonstrating that manifold decoupling is mandatory to preserve identifiable attack substructures in high-dimensional IoT environments.

Taken together, these geometric properties confirm that AMD-GAN generates distinct threat signatures while faithfully preserving the natural overlap of the underlying real distributions. This topological integrity is the fundamental prerequisite that enables both the SOC attribution transparency demonstrated in Section 6.2 and the downstream detection gains reported in Section 5.

7. Qualitative Comparison with State-of-the-Art Generative NIDS

To situate the AMD-GAN framework within the broader landscape of network security, this section provides a comparison with recent state-of-the-art (2024–2026) generative NIDS models. It is imperative to note that this is a qualitative and architectural comparison, rather than a controlled empirical head-to-head benchmark. Because the referenced models in the literature were evaluated under varying experimental conditions—including different data partitioning strategies, feature engineering pipelines, and downstream classifier configurations—their reported performance metrics are provided strictly for contextual reference. Consequently, rather than claiming direct empirical performance superiority over these external baselines, this section highlights how AMD-GAN addresses common architectural limitations through class-specific manifold isolation and adaptive cardinality regimes. Furthermore, for controlled empirical comparisons that isolate the architectural advantages of our proposal against standard monolithic cGANs and unregularized multi-generator arrays (representative of the TMG-GAN style approach), we direct the reader to the intrinsic fidelity evaluation in Section 5.5 and the geometric separability analysis in Section 6.3, which were executed under identical feature spaces and preprocessing conditions.

While the majority of works in Table 19 evaluate generative utility under a *Train-Real-Test-Real* (TRTR) paradigm, AMD-GAN is evaluated under the stricter *Train-Synthetic-Test-Real* (TSTR) protocol, where no real training data is available to the classifier. Under TSTR, the achievable performance ceiling is inherently lower than under TRTR, making direct numerical comparison of Macro F1-Score or Accuracy scores across rows methodologically unsound. The appropriate comparative axes are therefore distributional fidelity metrics (W_1, MMD^2), relative gains over each model's own imbalanced baseline, and operational viability constraints, dimensions where AMD-GAN establishes clear and verifiable advantages.

As demonstrated in Table 19, deep multimodal diffusion models such as MAGE-ID [26] and attention-based ensembles such as Transformer-GAN-AE [25] achieve topological realism under TRTR conditions. However, their iterative Markov chain denoising and quadratic spatial complexity $O(N^2)$ render them practically unscalable for real-time inference or on-demand adversarial auditing. Conversely, monolithic tabular approaches such as CTGAN [17] and shared-discriminator multi-generator models such as TMG-GAN [24] maintain lower latency but fail to fully decouple the optimization landscape across cardinality regimes, limiting their relative recovery impact on extremely sparse attack manifolds.

8. Limitations

Despite its demonstrated advantages, AMD-GAN presents some limitations that should be considered when deploying the framework in production environments. The adaptive training thresholds θ_{high} and θ_{low} are calibrated empirically for the evaluated feature dimensionalities and may require adjustment when applied to network environments with substantially different feature spaces or extraction paradigms. To enhance the framework's plug-and-play generalization across novel network topologies, exploring automated, data-driven calibration mechanisms represents a critical avenue for future work. Specifically, the integration of Bayesian Optimization during a brief warm-up phase,

Table 19: Qualitative comparison of recent generative NIDS frameworks. Performance metrics for external models are derived from their original publications.

Architecture	Datasets	Eval. Protocol	Intrinsic Fidelity (Distributional Quality)	Peak Classification Performance (reported)	Relative Lift over Imbalanced Baseline	Minority Class Recovery
CopulaGAN / CTGAN (2024) [17]	CIC-IDS2017, NSL-KDD	TRTR	Tuning-dependent divergence; no W_1 reported.	Inconsistent; degrades under severe data sparsity.	Marginal gains; limited on non-convex manifolds.	Limited improvement on disjoint or sparse clusters.
TMG-GAN (2024) [24]	CIC-IDS2017, UNSW-NB15	TRTR	Cosine Similarity penalty; no distributional distance reported.	CIC-IDS2017: 99.72% Macro F1, 99.63% Acc.	Consistent lifts vs. monolithic baselines.	Robust recovery; constrained by shared discriminator bottleneck.
CE-GAN (2025) [14]	UNSW-NB15, NSL-KDD	TRTR	No statistical distance metrics reported.	UNSW-NB15: 99.45% Accuracy.	Notable Macro-F1 gains in atypical minority classes.	Good; limited by unified generative channel.
Transf.-GAN-AE (2025) [25]	Edge-IIoTset, WUSTL-IIoT	TRTR	Bounded by Reconstruction Error; no W_1 reported.	Edge-IIoTset: 98.63% Acc., 99.53% AUC.	>98% Recall recovery for IIoT intrusions.	Exceptional; unscalable due to $O(N^2)$ attention complexity.
MAGE-ID (2026) [26]	CIC-IDS2017, NSL-KDD	TRTR	Advanced PRDC (Precision, Recall, Density, Coverage).	CIC-IDS2017: Macro-F1 ~0.964 (+1.9% over XGBoost).	Exceptional underrepresentation and coverage recovery.	Exceptional realism; prohibitive iterative Markov denoiser latency.
CTGSM-DNN (2025) [15]	CSE-CIC-IDS2018	TRTR	N/A; relies on external SMOTE-ENN post-filter.	CIC-IDS2018: 99.90% global Accuracy.	Modest; 80.0% Acc. on rare anomalies only.	Limited; base generator requires external heuristic correction.
AMD-GAN (Ours)	**CIC-IDS2017, UNSW-NB15, Edge-IIoTset**	**TSTR & TRTR**	**$W_1 = 0.009$, $MMD^2 < 0.027$** Excellent fidelity across all cardinality regimes.	**TRTR-Augmented Macro F1: CIC-IDS2017: 84.0% UNSW-NB15: 51.8% Edge-IIoTset: 93.0%** ***(See Table 12)***	**+10.1 pp Macro-F1**‡ (Syn-Balanced vs. Syn-Real, TSTR)	**Botnet: +45.2% F1 Web Attack: +22.9% F1 Peak VRAM: 1,575 MB; 90k flows in 15.7s.**

† AMD-GAN is evaluated under both TSTR (measuring pure synthetic generative fidelity) and TRTR protocols. The TRTR performance reported above strictly isolates the performance of AMD-GAN as a data augmentation technique against the original real-data baselines (see Section 5.3, Table 12). Direct numerical comparison of Accuracy/F1 scores across rows remains methodologically complex due to differing unaugmented real baselines in external works, but our dual-protocol reporting provides full methodological transparency.

‡ The reported +10.1 pp refers to the absolute improvement in Multiclass Macro F1-Score when training on class-balanced synthetic data versus class-imbalanced synthetic data, both under the TSTR protocol on CIC-IDS2017. This metric isolates the contribution of generative class balancing to minority attack detection, independently of real data availability.

or the use of unsupervised density estimators, could allow the system to dynamically infer these thresholds from the underlying data distribution without manual intervention.

The UNSW-NB15 results further evidence that the decoupled architecture is sensitive to extreme inter-class overlap: when class manifolds are geometrically proximate in the original feature space, isolated generators may capture distributions that introduce boundary ambiguity rather than resolve it, partially offsetting the benefits of manifold decoupling, as evidenced by the LGBM and KNN configurations on UNSW-NB15 (Table 12), where AMD-GAN ranks 5th and 2nd respectively. To mitigate this boundary ambiguity in environments with high inter-class overlap, a promising direction for future work is the integration of a hybrid regularization mechanism. By incorporating a supervised contrastive loss penalty across the parallel decoupled generators [35, 36], or introducing a margin-based boundary repulsion term during a brief joint fine-tuning phase, the framework could explicitly penalize the isolated models for synthesizing boundary-crossing samples. This would encourage the generators to actively maximize inter-class margins while retaining their primary focus on minority manifold fidelity.

From a computational standpoint, the cumulative training time of 4 hours and 47 minutes for a full from-scratch initialization represents a practical barrier for initial deployment. However, in edge environments requiring rapid model refresh in response to evolving threats, this cost is circumvented by AMD-GAN's decoupled architecture, which natively supports asynchronous incremental updates. If a novel attack vector is detected, the NIDS does not retrain the entire multi-class suite. Instead, a single class-specific generator can be instantiated or updated in strict isolation. Based on our hardware profiling (Table 5), training a new sparse-class generator from scratch on edge-compatible hardware requires approximately 45 to 60 minutes, based on projected scaling from our server-side profiling benchmarks. Furthermore, applying transfer learning (warm-starting) to fine-tune an existing generator with newly captured network telemetry reduces this update latency to an estimated 10–15 minutes. While the sequential strategy inherently prevents the joint optimization of inter-class boundaries, this modularity, combined with the strict 1,575 MB VRAM footprint, ensures that continuous online adaptation remains operationally viable in resource-constrained environments without prohibitive downtime.

Finally, a significant limitation of generative models operating on tabular flow features, as presented in this paper, is the difficulty of ensuring true semantic and causal validity. While our synthesis mixer employs exponential reversion, rounding, and bounding operations to successfully enforce basic numerical ranges, these range-based checks do not guarantee valid protocol semantics. For instance, the generative framework does not explicitly enforce valid combinations of TCP flags, semantic consistency between specific applications and transport ports, or the strict mathematical consistency that must exist between derived features such as flow duration, packet count, and bit rate. Consequently, while the synthetic traffic achieves high statistical fidelity and structural realism for downstream ML classification, it may still exhibit localized semantic inconsistencies that a rule-based Deep Packet Inspection (DPI) system could identify.

9. Conclusion

The structural imbalance inherent in modern network traffic presents a critical vulnerability for Machine Learning-based NIDS, systematically blinding classifiers to the rare intrusion vectors that pose the highest operational risk. In this paper, we demonstrated that standard monolithic generative architectures are structurally ill-equipped to address this, as majority-class gradient dominance inevitably forces conditional mode collapse. To overcome this, we introduced the Adaptive Manifold-Decoupled GAN (AMD-GAN), establishing that the class-wise architectural decoupling of features, combined with cardinality-aware optimization, provides a highly robust framework for high-fidelity minority attack synthesis.

By evaluating AMD-GAN across three heterogeneous network benchmarks, this study yielded several critical implications for both defensive hardening and offensive auditing. Defensively, the framework validates generative manifold decoupling as a competitive and often higher-ranked alternative to traditional interpolation techniques. Because AMD-GAN accurately maps non-convex, disjoint attack topologies without introducing boundary noise, it successfully rescues the detection capability of standard classifiers for critical, long-tail threats, drastically reducing the operational blind spots of the NIDS. Furthermore, the integration of explainable AI (SHAP and LIME) confirmed that these synthetic signatures maintain protocol-level semantics, enabling unambiguous traceability and transparent integration into practical Security Operations Center (SOC) workflows.

Offensively, AMD-GAN operationalizes as an automated Red Teaming engine that exposes a fundamental paradox in current NIDS evaluation paradigms: classifiers that achieve near-perfect metrics on standard held-out tests remain catastrophically vulnerable to targeted volumetric distributional shifts. By subjecting models to synthetic class-prior

shift stress scenarios, we demonstrated that structural robustness cannot be inferred from traditional static benchmarks alone, necessitating active adversarial auditing. Crucially, the decoupled sequential training architecture achieves these results with a minimal memory footprint, demonstrating that advanced generative augmentation and auditing are computationally viable directly on resource-constrained hardware.

Future work will explore the extension of this decoupled architecture to non-tabular data structures. To adapt the framework for raw packet payloads, the current MLP-based generators and critics would need to be replaced with sequence-to-sequence models or Transformer architectures capable of handling discrete byte streams, while strictly maintaining the class-specific manifold isolation. Similarly, graph-structured network topologies could be synthesized by integrating Graph Neural Networks (GNNs) into the isolated generative pipelines, preserving the decoupled training regime while capturing complex node communication patterns for Deep Packet Inspection (DPI) systems. Additionally, adapting the generator repository for Federated Learning environments represents a promising pathway for organizations to collaboratively train isolated attack manifolds without compromising privacy-sensitive network telemetry.

Acknowledgements

This publication is part of the project "CiberIA: Investigación e Innovación para la Integración de Ciberseguridad e Inteligencia Artificial" (Proyecto C079/23), financed by "European Union NextGeneration-EU, the Recovery Plan, Transformation and Resilience", through INCIBE. It has also been partially supported by the project SecAI (PID2022-139268OB-I00) funded by the Spanish Ministerio de Ciencia e Innovacion, and Agencia Estatal de Investigacion. Funding for open access charge: Universidad de Málaga / CBUA.

Conflict of Interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data and Code Availability

The datasets generated, models, source code implemented for AMD-GAN framework and tests performed are available here: https://doi.org/10.5281/zenodo.19212815.

References

[1] R. Ahsan, W. Shi, J. P. Corriveau, Network intrusion detection using machine learning approaches: Addressing data imbalance, IET Cyber-Physical Systems: Theory and Applications 7 (1) (2022) 30–39. doi:10.1049/cps2.12013.
URL https://doi.org/10.1049/cps2.12013

[2] H. A. Ahmed, A. Hameed, N. Z. Bawany, Network intrusion detection using oversampling technique and machine learning algorithms, PeerJ Computer Science 8 (2022) e820. doi:10.7717/PEERJ-CS.820.
URL https://peerj.com/articles/cs-820

[3] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, SMOTE: Synthetic Minority Over-sampling Technique, Journal of Artificial Intelligence Research 16 (2002) 321–357. doi:10.1613/jair.953.
URL http://dx.doi.org/10.1613/jair.953

[4] S. Bagui, K. Li, Resampling imbalanced data for network intrusion detection datasets, Journal of Big Data 2021 8:1 8 (1) (2021) 6–. doi:10.1186/s40537-020-00390-x.
URL https://link.springer.com/article/10.1186/s40537-020-00390-x

[5] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative Adversarial Networks, in: Proceedings of the International Conference on Neural Information Processing Systems (NIPS 2014), Vol. 3, Neural information processing systems foundation, 2014, pp. 2672–2680.
URL https://arxiv.org/pdf/1406.2661

[6] Tertsegha J. Anande, Mark S. Leeson, Generative Adversarial Networks (GANs): A Survey on Network Traffic Generation, International Journal of Machine Learning and Computing 12 (6) (11 2022). doi:10.18178/ijmlc.2022.12.6.1120.

[7] I. Sharafaldin, A. H. Lashkari, A. A. Ghorbani, Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization, in: Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP), 2018, pp. 108–116. doi:10.5220/0006639801080116.
URL https://doi.org/10.5220/0006639801080116

[8] N. Moustafa, J. Slay, UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set), in: Military Communications and Information Systems Conference (MilCIS), Canberra, Australia, 2015, pp. 1–6. doi:10.1109/milcis.2015.7348942.
URL https://doi.org/10.1109/milcis.2015.7348942

[9] M. A. Ferrag, O. Friha, D. Hamouda, L. Maglaras, H. Janicke, Edge-IIoTset: A New Comprehensive Realistic Cyber Security Dataset of IoT and IIoT Applications for Centralized and Federated Learning, IEEE Access 10 (2022) 40281–40306. doi:10.1109/ACCESS.2022.3165809.

[10] S. Lundberg, S.-I. Lee, A Unified Approach to Interpreting Model Predictions, Advances in Neural Information Processing Systems 2017-December (2017) 4766–4775.
URL http://arxiv.org/abs/1705.07874

[11] M. T. Ribeiro, S. Singh, C. Guestrin, "Why Should I Trust You?": Explaining the Predictions of Any Classifier, in: NAACL-HLT 2016 - 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Demonstrations Session, Association for Computational Linguistics (ACL), 2016, pp. 97–101. doi:10.18653/v1/n16-3020.
URL https://arxiv.org/pdf/1602.04938

[12] H. Ding, L. Chen, L. Dong, Z. Fu, X. Cui, Imbalanced data classification: A KNN and generative adversarial networks-based hybrid approach for intrusion detection, Future Generation Computer Systems 131 (7) (2022) 240–254. doi:10.1016/j.future.2022.01.026.
URL https://doi.org/10.1016/j.future.2022.01.026

[13] S. Huang, K. Lei, IGAN-IDS: An imbalanced generative adversarial network towards intrusion detection system in ad-hoc networks, Ad Hoc Networks 105 (6) (2020) 102177. doi:10.1016/j.adhoc.2020.102177.
URL https://doi.org/10.1016/j.adhoc.2020.102177

[14] Y. Yang, X. Liu, D. Wang, Q. Sui, C. Yang, H. Li, Y. Li, T. Luan, A CE-GAN based approach to address data imbalance in network intrusion detection systems, Scientific Reports 2025 15 (3 2025). doi:10.1038/s41598-025-90815-5.
URL https://www.nature.com/articles/s41598-025-90815-5

[15] S. Menssouri, E. M. Amhoud, A Conditional Tabular GAN-Enhanced Intrusion Detection System for Rare Attacks in IoT Networks, in: 2025 IEEE International Conference on Communications Workshops, ICC Workshops 2025, Institute of Electrical and Electronics Engineers Inc., 2025, pp. 1918–1923. doi:10.1109/ICCWorkshops67674.2025.11162182.
URL http://dx.doi.org/10.1109/ICCWorkshops67674.2025.11162182

[16] F. Kang, T. Feng, J. Lin, VAE-GAN-Guided Cross-Class Generation: A Class Imbalance Data Augmentation Method for Network Intrusion Detection, Electronics 2025, Vol. 14, Page 2103 14 (11) (2025) 2103. doi:10.3390/electronics14112103.
URL https://doi.org/10.3390/electronics14112103

[17] J. Wang, X. Yan, L. Liu, L. Li, Y. Yu, CTTGAN: Traffic Data Synthesizing Scheme Based on Conditional GAN, Sensors 2022, Vol. 22, Page 5243 22 (14) (2022) 5243. doi:10.3390/s22145243.
URL https://doi.org/10.3390/s22145243

[18] M. A. Rahman, G. A. Francia, H. Shahriar, Leveraging GANs for Synthetic Data Generation to Improve Intrusion Detection Systems, Journal of Future Artificial Intelligence and Technologies 1 (4) (2025) 429–439. doi:10.62411/faith.3048-3719-52.
URL https://faith.futuretechsci.org/index.php/FAITH/article/view/52

[19] J. H. Lee, K. H. Park, GAN-based imbalanced data intrusion detection system, Personal and Ubiquitous Computing 2019 25:1 25 (1) (2019) 121–128. doi:10.1007/s00779-019-01332-y.
URL https://link.springer.com/article/10.1007/s00779-019-01332-y

[20] M. Ring, D. Schlör, D. Landes, A. Hotho, Flow-based network traffic generation using Generative Adversarial Networks, Computers and Security 82 (2019) 156–172. doi:10.1016/j.cose.2018.12.012.
URL http://dx.doi.org/10.1016/j.cose.2018.12.012

[21] J. Yoon, D. Jarrett, M. Van Der Schaar, Time-series Generative Adversarial Networks, in: 33rd Conference on Neural Information Processing Systems, Vol. 32, 2019, pp. 5508–5518.

[22] A. Cheng, PAC-GAN: Packet Generation of Network Traffic using Generative Adversarial Networks, in: 2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference, IEMCON 2019, Institute of Electrical and Electronics Engineers Inc., 2019, pp. 728–734. doi:10.1109/IEMCON.2019.8936224.
URL https://ieeexplore.ieee.org/document/8936224

[23] A. Schoen, G. Blanc, P. F. Gimenez, Y. Han, F. Majorczyk, L. Me, A Tale of Two Methods: Unveiling the Limitations of GAN and the Rise of Bayesian Networks for Synthetic Network Traffic Generation, in: Proceedings - 9th IEEE European Symposium on Security and Privacy Workshops, Euro S and PW 2024, Institute of Electrical and Electronics Engineers Inc., 2024, pp. 273–286. doi:10.1109/EuroSPW61312.2024.00036.
URL https://ieeexplore.ieee.org/document/10628855

[24] H. Ding, Y. Sun, N. Huang, Z. Shen, X. Cui, TMG-GAN: Generative Adversarial Networks-Based Imbalanced Learning for Network Intrusion Detection, IEEE Transactions on Information Forensics and Security 19 (2024) 1156–1167. doi:10.1109/TIFS.2023.3331240.
URL https://ieeexplore.ieee.org/document/10312801

[25] A. Salehiyan, P. S. Moghaddam, M. Kaveh, An Optimized Transformer–GAN–AE for Intrusion Detection in Edge and IIoT Systems : Experimental Insights from WUSTL-IIoT-2021, EdgeIIoTset, and TON_IoT Datasets, Future Internet 17 (7) (2025) 1–34. doi:10.3390/fi17070279.
URL https://research.aalto.fi/en/publications/an-optimized-transformerganae-for-intrusion-detection-in-edge-and/

[26] M. A. Loodaricheh, M. H. Manshaei, A. Raja, MAGE-ID: A Multimodal Generative Framework for Intrusion Detection Systems, in: 2026 International Conference on Computing, Networking and Communications (ICNC), IEEE Computer Society, Maui, USA, 2026, pp. 585–589. doi:10.1109/ICNC68183.2026.11416962.
URL https://doi.org/10.1109/ICNC68183.2026.11416962

[27] A. Awasthi, A. K. Prajapati, P. Vediya, R. B. Battula, TICNN - A Hybrid Light-Weight CNN for Large Imbalanced Multi-Class Datasets on Intrusion Detection System in IoT, IEEE Access (2026). doi:10.1109/ACCESS.2026.3663379.
URL https://ieeexplore.ieee.org/document/11389773

[28] T. T. Cai, C. H. Zhang, H. H. Zhou, Optimal rates of convergence for covariance matrix estimation, Annals of Statistics 38 (4) (2010) 2118–2144. doi:10.1214/09-AOS752.

[29] M. Arjovsky, S. Chintala, L. Bottou, Wasserstein Generative Adversarial Networks, in: Proceedings of Machine Learning Research, PMLR, 2017, pp. 214–223.
URL https://proceedings.mlr.press/v70/arjovsky17a.html

[30] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, A. C. Courville, Improved Training of Wasserstein GANs, Advances in Neural Information Processing Systems 30 (2017).
URL https://github.com/igul222/improved

[31] Canadian Institute for Cybersecurity, CICFlowMeter (2017).
URL https://github.com/CanadianInstituteForCybersecurity/CICFlowMeter

[32] P. Sanchez-Serrano, R. Rios, I. Agudo, A decision framework for privacy-preserving synthetic data generation, Computers and Electrical Engineering 126 (2025) 110468. doi:10.1016/J.COMPELECENG.2025.110468.
URL https://www.sciencedirect.com/science/article/pii/S0045790625004112

[33] J. Demšar, Statistical Comparisons of Classifiers over Multiple Data Sets, Journal of Machine Learning Research 7 (2006) 1–30.

[34] L. v. d. Maaten, G. Hinton, Visualizing Data using t-SNE, Journal of Machine Learning Research 9 (86) (2008) 2579–2605.
URL http://jmlr.org/papers/v9/vandermaaten08a.html

[35] P. Khosla, P. Teterwak, C. Wang, A. Sarna, G. Research, Y. Tian, P. Isola, A. Maschinot, C. Liu, D. Krishnan, Supervised Contrastive Learning, in: Proceedings of the 34th International Conference on Neural Information Processing Systems, Vancouver, Canada, 2020, pp. 18661 – 18673.
URL https://dl.acm.org/doi/abs/10.5555/3495724.3497291

[36] M. Kang, J. Park, ContraGAN: Contrastive Learning for Conditional Image Generation, in: Proceedings of the 34th International Conference on Neural Information Processing Systems, Vancouver, Canada, 2020, pp. 21357 –21369.
URL https://dl.acm.org/doi/10.5555/3495724.3497517