future internet

MDPI

Article

Mitigating Class Imbalance in Network Intrusion Detection with Feature-Regularized GANs

Jing Li, Wei Zong *, Yang-Wai Chow * and Willy Susilo

Institute of Cybersecurity and Cryptology, School of Computing and Information Technology, University of Wollongong, Wollongong, NSW 2522, Australia; jl571@uowmail.edu.au (J.L.); wsusilo@uow.edu.au (W.S.)
* Correspondence: wzong@uow.edu.au (W.Z.); caseyc@uow.edu.au (Y.-W.C.)

Abstract: Network Intrusion Detection Systems (NIDS) often suffer from severe class imbalance, where minority attack types are underrepresented, leading to degraded detection performance. To address this challenge, we propose a novel augmentation framework that integrates Soft Nearest Neighbor Loss (SNNL) into Generative Adversarial Networks (GANs), including WGAN, CWGAN, and WGAN-GP. Unlike traditional oversampling methods (e.g., SMOTE, ADASYN), our approach improves feature-space alignment between real and synthetic samples, enhancing classifier generalization on rare classes. Experiments on NSL-KDD, CSE-CIC-IDS2017, and CSE-CIC-IDS2018 show that SNNL-augmented GANs consistently improve minority-class F1-scores without degrading overall accuracy or majority-class performance. UMAP visualizations confirm that SNNL produces more compact and class-consistent sample distributions. We also evaluate the computational overhead, finding the added cost moderate. These results demonstrate the effectiveness and practicality of SNNL as a general enhancement for GAN-based data augmentation in imbalanced NIDS tasks.

Keywords: network intrusion detection; WGAN; minority-class synthesis; soft nearest neighbor loss; data augmentation

Academic Editors: Olusola Tolulope Odeyomi and Temitayo Olowu

Received: 31 March 2025
Revised: 8 May 2025
Accepted: 12 May 2025
Published: 13 May 2025

Citation: Li, J.; Zong, W.; Chow, Y.-W.; Susilo, W. Mitigating Class Imbalance in Network Intrusion Detection with Feature-Regularized GANs. *Future Internet* **2025**, *17*, 216. https://doi.org/10.3390/fi17050216

1. Introduction

Network Intrusion Detection Systems (NIDS) serve as a fundamental defence mechanism against evolving cyber threats by monitoring and analysing network traffic for malicious activities. In real-world settings—such as industrial control systems (ICS), smart grids, and medical networks—even a single undetected stealthy intrusion, such as infiltration or privilege escalation, can result in persistent compromise, lateral movement, or data exfiltration, ultimately posing severe operational or safety risks. These scenarios highlight the importance of accurate and robust intrusion detection.

NIDS approaches can be broadly categorized into signature-based and anomaly-based detection. While signature-based methods rely on predefined attack signatures, they struggle against novel and zero-day threats. Anomaly-based detection, powered by machine learning models, offers greater adaptability but suffers from a major challenge: class imbalance in intrusion detection datasets. Specifically, minority attack classes—such as User-to-Root (U2R), Remote-to-Local (R2L), infiltration, and Web Attacks—are significantly underrepresented, resulting in biased classifiers with poor recall and F1-scores on critical but infrequent attack categories, severely limiting real-world detection performance. Prior studies [1] emphasize that visual separation between minority and majority classes in feature space is often blurred, further complicating detection under imbalanced settings.

To alleviate class imbalance, Generative Adversarial Networks (GANs) have been increasingly adopted for synthetic data augmentation. Unlike traditional oversampling techniques such as Synthetic Minority Over-sampling Technique (SMOTE), Adaptive Synthetic Sampling (ADASYN), and Random Oversampler (ROS), GANs generate synthetic samples that better capture the underlying data distribution. Among GAN variants, Wasserstein GAN (WGAN), Conditional WGAN (CWGAN), and WGAN-GP have demonstrated superior stability and sample quality in NIDS applications. However, existing GAN-based augmentation methods still suffer from two critical limitations:

- **Feature Misalignment:** The synthetic samples generated by conventional GANs often fail to preserve key feature characteristics of real network traffic, leading to suboptimal classification performance when integrated into training data.
- **Overfitting:** While some GAN-based techniques focus on improving sample realism, they may inadvertently reduce sample diversity, limiting their generalization to new attack variations.

To address these limitations, we propose a novel method that incorporates Soft Nearest Neighbor Loss (SNNL) into GAN training to explicitly enforce intra-class feature compactness and inter-class separation. Unlike prior approaches that modify generator architecture or use adversarial loss alone, our framework enhances WGAN, CWGAN, and WGAN-GP, thereby improving both the feature alignment and discriminative utility of generated minority-class samples.

We validate the effectiveness of our method on three widely used benchmark datasets: NSL-KDD, CSE-CIC-IDS2017, and CSE-CIC-IDS2018. Extensive experiments demonstrate that SNNL-enhanced GANs significantly improve minority-class detection without degrading majority-class performance or overall accuracy. Furthermore, we analyse the computational overhead introduced by SNNL, showing that the increase in training cost remains moderate.

The key contributions of this study are summarized as follows:

- We propose a novel method to exploit SNNL as an adaptive feature alignment loss to enhance GAN-generated samples by reducing feature misalignment between synthetic and real network traffic.
- We integrate SNNL into different GAN frameworks, e.g., WGAN, CWGAN and WGAN-GP, demonstrating its generalizability across different GAN architectures and datasets.
- We empirically show that our method surpasses traditional oversampling methods (SMOTE, ADASYN, ROS) and GAN-based augmentation without SNNL on widely used benchmark datasets: NSL-KDD, CSE-CIC-IDS2017 and CSE-CIC-IDS2018.
- We demonstrate via visualization that SNNL enhances GAN frameworks to produce more realistic and classifier-friendly synthetic samples, leading to substantial improvements in minority-class F1-scores.

The remainder of this paper is structured as follows: Section 2 provides a review of GAN-based data augmentation for NIDS, highlighting the limitations of existing augmentation strategies. Section 3 details the proposed SNNL-enhanced GAN frameworks. Section 4 presents our experimental setup and results. Section 5 analyses the effectiveness of SNNL via visualization. Section 6 discusses the remaining challenge. Finally, Section 7 concludes this paper.

2. Related Work

Handling class imbalance in Intrusion Detection Systems (IDS) has been a persistent challenge in both traditional and deep learning-based approaches. Early studies

primarily focused on data-level solutions, such as oversampling and two-stage classification pipelines. For instance, Zong et al. [2] proposed a two-stage classifier that combines SMOTE with ensemble learning, improving the detection of rare attacks through stage-wise specialization. These traditional methods, however, often suffer from overfitting or poor feature generalization.

More recently, Generative Adversarial Networks (GANs) have emerged as a promising approach for Intrusion Detection Systems (IDS), particularly in addressing data imbalance and generating high-quality synthetic attack samples. Existing research on GAN-based IDS can be categorized into two main directions: (1) applying GANs for minority-class data augmentation without significant modifications to their loss functions, and (2) modifying the GAN framework, including loss functions and architectural adjustments, to enhance training stability, sample quality, and minority-class detection. In this section, we review relevant studies in both categories, focusing on their contributions and limitations in handling class imbalance.

2.1. GAN-Based Intrusion Detection Without Major Loss Modifications

Many studies have employed standard GAN or Wasserstein GAN (WGAN) frameworks to improve intrusion detection by oversampling rare attack classes. For instance, Lee and Park [3] utilized a GAN-based approach to generate synthetic samples for underrepresented attack types in the CIC-IDS-2017 dataset, successfully mitigating class imbalance. Similarly, Ding et al. [4] proposed a hybrid IDS model integrating GANs with Deep Autoencoders (DAE) and Random Forest (RF), enhancing the detection of rare attack patterns in 5G IoT environments.

Other works have explored integrating GANs with classification models to enhance feature representation learning. Liao et al. [5] introduced an auxiliary classifier into the GAN framework, ensuring that the generated samples not only resemble real network traffic but also preserve distinctive attack characteristics. Araujo-Filho et al. [6] leveraged WGAN with Temporal Convolutional Networks (TCN) and self-attention mechanisms to improve the efficiency of sequence modelling in IDS.

Several studies have focused on balancing the quality and diversity of generated samples. Liu et al. [7] incorporated WGAN-GP with ANOVA feature selection to reduce high-dimensional feature complexity while effectively generating minority-class samples. Li et al. [8] introduced CL-GAN, a semi-supervised architecture that integrates Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks, improving detection accuracy for DoS attacks in IoT environments.

To enhance scalability, Poongodi and Hamdi [9] proposed a distributed multi-discriminator GAN, reducing computational overhead in large-scale IDS deployments. Rahman et al. [10] introduced a novel termination criterion based on box-plot distribution matching, ensuring that the generated data closely aligns with real-world traffic. Furthermore, Li et al. [11] and Sophia et al. [12] demonstrated that augmenting IDS datasets using GAN-generated samples significantly improves classification accuracy when integrated with deep learning models such as CNN, BiLSTM, or LightGBM.

2.2. GANs with Modified Loss Functions or Network Architectures

While many studies adopt standard GAN frameworks, others have sought to enhance sample quality and training stability by modifying GAN loss functions or network architectures. Chen [13] proposed a BiGAN that replaces Jensen–Shannon (JS) divergence with Wasserstein distance, incorporating feature matching and residual loss to mitigate mode collapse and improve sample realism. Xu et al. [14] further refined BiGAN by introducing

label flipping and a one-class classifier, optimizing the generator's ability to reconstruct and synthesize rare attack samples.

Some studies have focused on multi-task learning within GAN architectures. Ding et al. [15] introduced TMG-GAN, a multi-generator WGAN where each generator specializes in different attack types, while the discriminator performs both real/fake discrimination and multi-class classification using cosine similarity loss and spectral normalization. Boppana and Bagade [16] proposed GAN-AE, an autoencoder-based GAN architecture that leverages reconstruction error as an anomaly detection mechanism. Similarly, Peng et al. [17] employed a BiGAN with a denoising generator to detect and restore adversarial examples in IDS.

Optimizing decision boundaries has been another key research area. Ahmad et al. [18] developed a Boundary-Seeking GAN (BGAN), encouraging the generator to produce attack samples near decision boundaries, thereby improving minority-class detection. To handle the complexity of time-dependent traffic data, Feng et al. [19] replaced traditional CNN-based generators with Transformer-based architectures, combining residual and discriminative loss functions.

Other studies have explored adaptive GAN training mechanisms. Bai et al. [20] proposed SC-WGAN, where the generator receives synchronous feedback from both the classifier and discriminator, explicitly improving the quality of synthetic samples for underrepresented attack types. Zhang et al. [21] introduced an MLP encoder and U-Net generator to capture fine-grained traffic patterns, enhancing anomaly detection. Finally, Strickland et al. [22] combined CopulaGAN with Deep Reinforcement Learning (DRL), optimizing both binary and multi-class intrusion detection through a reward-based mechanism.

3. Proposed Methodology

While prior research has significantly advanced GAN-based data augmentation for IDS, challenges remain in stabilizing training, ensuring realistic feature representations, and improving the alignment between synthetic and real minority-class samples. Most existing approaches rely on either basic data augmentation or architectural modifications to enhance sample diversity. There is limited focus on optimizing the feature space alignment between real and synthetic data.

To address this gap, we propose incorporating SNNL into GAN frameworks. Unlike conventional GAN-based augmentation techniques, which primarily focus on generating more data, SNNL enhances the quality of synthetic samples by enforcing intra-class compactness and inter-class separation in the feature space. This improves the classifier's ability to distinguish minority-class attack samples while maintaining stable performance on majority classes.

In the following sections, we detail the GAN architecture we focus on, the integration of SNNL into the GAN training process, and the loss functions designed.

3.1. GAN Architecture

In this work, we employ Wasserstein GAN (WGAN), Conditional WGAN (CWGAN), and WGAN with gradient penalty (WGAN-GP) as the underlying generative models due to their superior stability and effectiveness in data augmentation. WGAN improves upon standard GANs by replacing the Jensen–Shannon divergence with the Wasserstein distance, which leads to more stable training and mitigates mode collapse. CWGAN further enhances WGAN by conditioning both the generator and the critic on class labels, enabling the generation of targeted attack samples. WGAN-GP builds on WGAN by introducing a gradient penalty to enforce the Lipschitz constraint, offering improved convergence without relying on weight clipping.

Despite these architectural advancements, WGAN, CWGAN, and WGAN-GP all lack explicit mechanisms to enforce feature-level alignment between real and synthetic data. This limitation motivates the incorporation of Soft Nearest Neighbor Loss (SNNL), which refines the feature representations of synthetic samples and ensures better alignment with the distribution of real network traffic.

3.2. Soft Nearest Neighbor Loss (SNNL)

The primary limitation of conventional GAN-based augmentation lies in its inability to explicitly optimize the feature similarity between synthetic and real samples. As a result, classifiers trained on GAN-augmented datasets may still struggle with the minority-class detection due to the feature space discrepancy. To address this, we introduce Soft Nearest Neighbor Loss (SNNL) [23] into the GAN training process, ensuring that generated samples not only resemble real data but also align well in the feature space.

Mathematically, given a dataset with feature embeddings $X = \{x_1, x_2, \ldots, x_N\}$ and corresponding class labels $Y = \{y_1, y_2, \ldots, y_N\}$, SNNL is defined as

$$SNNL(X, Y, T) = -\frac{1}{N}\sum_{i=1}^{N} \log \left(\frac{\sum\limits_{\substack{j=1 \\ j\neq i, y_i = y_j}}^{N} e^{-\frac{\|x_i - x_j\|^2}{T}}}{\sum\limits_{\substack{k=1 \\ k\neq i}}^{N} e^{-\frac{\|x_i - x_k\|^2}{T}}} \right) \quad (1)$$

where $\|x_i - x_j\|^2$ represents the squared Euclidean distance between feature vectors, and T is a temperature parameter that controls the weighting of distances. Lower values of T emphasize closer neighbors, reinforcing tighter intra-class clustering, whereas higher values allow for broader variations in feature space.

By minimizing SNNL, the generator is encouraged to produce synthetic samples that exhibit stronger intra-class cohesion while maintaining clear inter-class separability. This ensures that the generated samples not only resemble real data but also preserve the semantic structure of network traffic in the feature space. Furthermore, SNNL helps mitigate mode collapse by enforcing sample diversity, thereby improving the generalizability of the trained IDS classifier.

3.3. Comparison with Other Feature Alignment Losses

While various feature alignment losses have been proposed to improve representation learning—such as Triplet Loss [24] and Center Loss [25]—we adopt Soft Nearest Neighbor Loss (SNNL) [23] due to its effectiveness and compatibility in the GAN-based data augmentation setting, especially under class imbalance.

Triplet Loss encourages a margin between intra-class and inter-class distances but requires constructing informative triplets of anchor, positive, and negative samples. In imbalanced datasets, especially where minority-class samples are scarce, reliable triplet construction becomes difficult. Furthermore, it relies on complex mining strategies such as hard negative sampling, which significantly increases training complexity and may introduce instability [26].

Center Loss minimizes the intra-class variance by pulling features toward their respective class centers. However, the effectiveness of Center Loss depends on stable center estimation, which is challenging when the number of minority-class samples is limited. In addition, Center Loss is typically used alongside a classification loss (e.g., softmax), making it less directly applicable to unsupervised GAN training [25].

By contrast, SNNL is a non-parametric and fully differentiable loss that encourages intra-class cohesion and inter-class dispersion through pairwise distance-based relationships in the embedding space. It avoids the need for explicit triplet construction or center computation and integrates seamlessly into adversarial learning. Prior work [23] also demonstrates that SNNL improves both interpretability and generalization in representation learning, making it a natural fit for GAN-based augmentation where better feature alignment of synthetic samples is crucial.

3.4. Integration of SNNL into WGAN Training

While WGAN improves training stability over standard GANs by addressing issues related to vanishing gradients, it does not inherently enforce feature-level similarity between real and generated samples. Consequently, synthetic data generated by WGAN may still suffer from feature misalignment, limiting its effectiveness in minority-class augmentation. To overcome this limitation, we incorporate SNNL as an additional regularization term into the critic loss during WGAN training.

The training objective of WGAN consists of optimizing the critic to approximate the Wasserstein distance between real and generated data distributions. Unlike standard GANs, which use a discriminator for binary classification, WGAN replaces it with a critic $C(x)$ that directly estimates the difference in expected values between real and generated samples. The original critic loss in WGAN is defined as

$$\mathcal{L}_{\text{WGAN}} = \mathbb{E}_{x \sim p_{\text{real}}}[C(x)] - \mathbb{E}_{\hat{x} \sim p_{\text{gen}}}[C(\hat{x})] \tag{2}$$

where $C(x)$ is the critic's score for a real sample x, and $C(\hat{x})$ is the score for a generated sample $\hat{x}$. The critic is trained to minimize the negative of this objective, thereby approximating the Wasserstein distance between the two distributions. To ensure convergence, the critic is updated multiple times per generator update, and weight clipping is applied to satisfy the Lipschitz constraint.

To promote feature-level alignment, we extend the critic's loss by introducing a Soft Nearest Neighbor Loss term applied to intermediate features of real samples:

$$\mathcal{L}_{\text{WGAN+SNNL}} = -\mathcal{L}_{\text{WGAN}} + \lambda \cdot \mathcal{L}_{\text{SNNL}} \tag{3}$$

where λ is a regularization coefficient balancing adversarial and alignment objectives. This combined loss is minimized during critic training. Importantly, SNNL is computed on all training data and does not alter the generator loss. As the critic learns more structured and class-aware representations, the generator indirectly benefits through improved training signals.

To compute SNNL, we extract feature embeddings from the intermediate layers of the critic. Given a batch of real samples $\{x_1, x_2, \ldots, x_N\}$, their feature embeddings are defined as

$$X = \{f(x_1), f(x_2), \ldots, f(x_N)\} \tag{4}$$

where $f(x)$ denotes the activation of an intermediate layer in the critic. Applying SNNL in this feature space encourages intra-class compactness and inter-class separation, improving generalization when incorporating synthetic samples into downstream classifiers.

3.5. Training Procedure

The training process follows a two-step optimization strategy. The critic is first updated to estimate the Wasserstein distance while incorporating SNNL to improve feature space alignment. Once the critic stabilizes, the generator is updated to minimize both the WGAN loss and SNNL regularization. The complete procedure is outlined in Algorithm 1.

Algorithm 1 Training procedure for WGAN+SNNL

Input: initialized generator G; initialized critic C; critic update frequency $freq$.
Output: well-trained generator G; well-trained critic C.
for each training iteration **do**
  Sample a mini-batch of real data $x \sim p_{\text{real}}$.
  Generate synthetic samples $\hat{x} = G(z)$, where $z \sim p(z)$.
  Compute critic loss $\mathcal{L}_{\text{WGAN}}$.
  Extract feature representations $f(x)$ and $f(\hat{x})$.
  Compute SNNL regularization $\mathcal{L}_{\text{SNNL}}$.
  Update critic C using $\mathcal{L}_{\text{WGAN+SNNL}}$.
  if iteration $\mod freq = 0$ **then**
    Update generator G using gradients from $\mathcal{L}_{\text{WGAN+SNNL}}$.
  end if
  Apply weight clipping to critic C.
end for

4. Experimental Results

4.1. Setup

We conduct experiments on three widely used benchmark datasets: NSL-KDD, CSE-CIC-IDS2017, and CSE-CIC-IDS2018. All datasets are used in their processed form, i.e., as tabular features extracted from raw network traffic, rather than raw packets. This ensures compatibility with feature-based learning models and aligns with common practice in prior NIDS studies. Details of these datasets and how to preprocess them are presented in the following sections.

4.1.1. NSL-KDD Dataset

NSL-KDD [27] is an improved version of the KDD99 dataset, designed to address redundancy and imbalance issues. It consists of four major attack categories: Denial of Service (DoS), Probe, Remote-to-Local (R2L), and User-to-Root (U2R). Among these, U2R and R2L are significantly underrepresented, making them the primary focus for evaluating the effectiveness of minority-class augmentation.

Dataset Splitting: We directly adopt the original training and testing split provided in NSL-KDD without additional modifications. This ensures that our results remain comparable with prior studies.

4.1.2. CSE-CIC-IDS2017 Dataset

CSE-CIC-IDS2017 [28] contains realistic network traffic and diverse attack scenarios, including Brute Force, Botnet, PortScan, DoS, Web Attack, and infiltration. Among them, Bot and infiltration classes are highly imbalanced, and thus serve as the primary targets for augmentation.

Dataset Splitting: We split the dataset into 70% for training and 30% for testing. This fixed partition is used consistently across all experiments to ensure fair and reproducible evaluation.

4.1.3. CSE-CIC-IDS2018 Dataset

CSE-CIC-IDS2018 [28] provides realistic attack scenarios, including DoS, DDoS, Brute Force, Botnet, Web Attacks, and infiltration. However, certain attack categories, such as infiltration and Web Attacks, suffer from severe data imbalance, making them the key targets for augmentation.

Dataset Splitting: Given the large-scale nature of CSE-CIC-IDS2018, we apply a two-stage stratified sampling strategy:

- First, we extract 10% of the entire dataset, ensuring that the class distribution remains proportional to the original dataset.
- Then, we allocate 70% of this subset for training and 30% for testing.

This strategy maintains the dataset's class imbalance while significantly reducing computational overhead.

4.1.4. Data Preprocessing

We apply consistent preprocessing to all datasets.

First, we apply Least Absolute Shrinkage and Selection Operator (Lasso) feature selection to identify the most relevant attributes, removing redundant or low-impact features. This step reduces input dimensionality, thereby improving training efficiency and stability across all GAN variants.

Categorical features (e.g., protocol type, service, flag) are one-hot encoded, while continuous numerical features are normalized to the $[0, 1]$ range using Min–Max scaling. Rows with missing values are discarded to preserve data integrity.

To train WGAN, CWGAN, and WGAN-GP, we adopt a two-stage augmentation pipeline. We first apply SMOTE to expand each minority class to the size of the majority class. This SMOTE-resampled dataset is then used to train the respective GAN model. Once training converges, the GAN is used to generate additional synthetic minority-class samples. These generated samples are then combined with the original training data to construct the final dataset used for classification.

For CWGAN, class labels are additionally used to guide generation, enabling label-consistent augmentation. WGAN-GP follows the same pipeline as WGAN but replaces weight clipping with a gradient penalty for improved training stability.

For comparison, traditional oversampling methods—SMOTE, ADASYN, and ROS—are applied directly to the original training set.

4.1.5. Classification Methods

To evaluate the effectiveness of the proposed augmentation strategies, we employ three widely used classifiers: Random Forest (RF), K-Nearest Neighbors (KNN), and XGBoost. For each experiment, we generate synthetic samples once and apply all three classifiers independently to assess performance. This process is repeated three times under consistent experimental conditions, and the final results are reported as the average across all runs and classifiers. This setup ensures robustness and minimizes variance caused by any single classifier or run.

4.1.6. Training Configuration

We employ WGAN and CWGAN with SNNL for generating synthetic minority-class samples. In a similar manner to other work, both the generator and critic architectures use fully connected layers with LeakyReLU activation and batch normalization.

The detailed training procedure is as follows:

- **Optimizer:** RMSProp with a learning rate of 10^{-4}.
- **Batch Size:** 256.
- **Training Duration:** 500 epochs.
- **WGAN Critic Constraint:** Weight clipping in the range $[-0.01, 0.01]$.
- **SNNL Temperature Parameter:** $T = 0.1$, selected via grid search (see Section 4.1.7).

4.1.7. SNNL Temperature Selection

To determine an appropriate value for the SNNL temperature parameter T, we performed a sensitivity analysis using a validation-based approach. Specifically, we split the

original NSL-KDD training set into 90% for training and 10% for validation. For each value $T \in \{0.01, 0.1, 0.3, 0.5, 1\}$, we generated synthetic samples using WGAN+SNNL and trained both Random Forest and XGBoost classifiers. The performance was evaluated on the held-out validation set, and each configuration was repeated three times to ensure robustness. The average results are summarized in Table 1.

Table 1. SNNL temperature sensitivity on NSL-KDD dataset (validation set, averaged over 3 runs).

Model	Temp	Accuracy	Macro-F1	FPR (%)	Detection Rate (%)
Random Forest	0.01	1.00	0.94	0.14	93.74
XGBoost	0.01	1.00	0.98	0.08	99.82
Random Forest	0.1	1.00	0.94	0.14	93.96
XGBoost	0.1	1.00	**0.97**	**0.03**	**99.97**
Random Forest	0.3	1.00	0.94	0.15	93.87
XGBoost	0.3	1.00	0.97	0.07	99.92
Random Forest	0.5	1.00	0.94	0.17	93.61
XGBoost	0.5	1.00	0.98	0.05	99.93
Random Forest	1.0	1.00	0.94	0.16	93.81
XGBoost	1.0	1.00	0.97	0.06	98.09

As seen in Table 1, all temperature settings yielded high classification accuracy, but $T = 0.1$ provided the best overall trade-off across multiple metrics. Notably, XGBoost achieved the highest macro-F1 score (0.97), the lowest false positive rate (0.03%), and the highest detection rate (99.97%) at $T = 0.1$. Random Forest results remained stable across different temperatures, suggesting robustness. Based on these observations, we fix $T = 0.1$ in all subsequent experiments across all datasets.

4.1.8. Computational Resources

All experiments were conducted on a high-performance workstation equipped with the following:

- **CPU:** Intel Core i7-14700KF @ 3.4GHz (Turbo Boost up to 5.6 GHz).
- **GPU:** NVIDIA RTX 4070 Ti OC (12GB GDDR6X VRAM).
- **Memory:** 32GB DDR5-6400 RAM.

PyTorch 1.13.1 with CUDA 11.7 was utilized to leverage GPU acceleration. Mixed-precision computation was employed to enhance memory efficiency and training speed.

4.2. NSL-KDD Results

4.2.1. Performance on Minority Class

Table 2 presents the precision, recall, and F1-score comparisons for the minority classes (U2R and R2L) across traditional oversampling techniques (SMOTE, ADASYN, ROS) and several GAN variants (WGAN, CWGAN, WGAN-GP) along with their SNNL-enhanced counterparts. The best results are highlighted in **bold**, and the second-best are underlined.

Traditional oversampling methods (SMOTE, ADASYN, ROS) generally yield lower performance in minority-class detection compared to GAN-based methods. For R2L, SMOTE achieves the best performance among traditional baselines with an F1-score of 0.150, while ADASYN and ROS follow with 0.137 and 0.107, respectively. A similar pattern is observed for U2R, where SMOTE and ADASYN reach 0.150 and 0.137, and ROS again performs the worst at 0.107.

Table 2. Minority-class performance comparison on NSL-KDD dataset.

Method	R2L			U2R		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score
SMOTE	0.930	0.097	0.150	0.477	0.097	0.150
ADASYN	0.910	0.073	0.137	0.497	0.090	0.137
ROS	0.887	0.057	0.107	0.423	0.070	0.107
WGAN	0.980	0.106	0.168	0.790	0.090	0.154
WGAN+SNNL	0.980	0.110	0.200	0.795	0.095	0.170
CWGAN	0.980	0.113	0.200	0.732	0.090	0.155
CWGAN+SNNL	**0.983**	0.112	0.200	**0.817**	**0.102**	**0.175**
WGAN-GP	0.980	0.090	0.167	0.697	0.080	0.137
WGAN-GP+SNNL	0.979	**0.120**	**0.217**	0.767	0.092	0.154

In contrast, GAN-based methods demonstrate noticeable improvements over traditional baselines. For R2L, WGAN and CWGAN achieve F1-scores of 0.168 and 0.200, while WGAN-GP achieves 0.167. Notably, integrating SNNL into WGAN-GP raises the F1-score to **0.217**, the highest among all evaluated methods. For U2R, CWGAN+SNNL outperforms all baselines with an F1-score of **0.175**, followed by WGAN+SNNL (0.170) and WGAN-GP+SNNL (0.154), each exceeding their corresponding base models.

These improvements may be attributed to SNNL's effect on improving either precision, recall, or both. For instance, WGAN-GP+SNNL raises recall from 0.090 to 0.120 for R2L while maintaining competitive precision. Although a slight decrease in recall is observed for CWGAN on U2R after SNNL integration, the overall F1-score still improves.

The observed performance gains across different GAN variants suggest the potential effectiveness of the proposed SNNL approach in addressing class imbalance and improving detection of underrepresented attack categories.

4.2.2. Overall Accuracy and Macro-F1

Table 3 presents the overall classification accuracy and macro-F1 scores on the NSL-KDD dataset. Overall accuracy differences among all methods are relatively small, with CWGAN+SNNL achieving the highest value at 76.8%. Other GAN-based methods such as WGAN+SNNL and WGAN-GP+SNNL also show slight improvements over their respective baselines, indicating that the inclusion of SNNL does not negatively impact general classification performance.

Table 3. Overall accuracy and macro-F1 comparison on NSL-KDD dataset.

Method	Accuracy (%)	Macro-F1 (%)
SMOTE	76.3	53.5
ADASYN	76.5	53.8
ROS	75.6	50.9
WGAN	76.5	54.2
WGAN+SNNL	76.7	55.1
CWGAN	76.7	54.8
CWGAN+SNNL	76.8	55.3
WGAN-GP	76.0	53.4
WGAN-GP+SNNL	76.7	56.7

Compared to traditional oversampling methods (SMOTE, ADASYN, ROS), GAN-based approaches yield slightly higher macro-F1 scores, which reflect better class-wise balance. Notably, WGAN-GP+SNNL achieves the highest macro-F1 of 56.7%, an increase of 3.3 percentage points over its base model, WGAN-GP. These results suggest that SNNL

may contribute to improving the discriminative utility of generated samples across all classes, particularly in imbalanced settings.

4.2.3. Majority-Class Performance

Figure 1 illustrates the F1-score comparison for majority classes (DoS, Normal, and Probe) across traditional oversampling methods and various GAN-based augmentation strategies, including SNNL-enhanced variants.

Figure 1. F1-score comparison by majority class on NSL-KDD dataset.

For DoS and Normal classes, all methods—both traditional and GAN-based—achieve consistently high F1-scores (approximately 87–89% for DoS and 79% for Normal), indicating these classes are well represented and reliably detected regardless of the augmentation strategy.

In the Probe category, more noticeable differences are observed. Traditional methods such as SMOTE and ADASYN achieve F1-scores of 73% and 75%, respectively, while ROS shows the lowest performance at 68%. Most GAN variants (WGAN, CWGAN, and their SNNL-enhanced versions) maintain F1-scores around 70–71%. However, WGAN-GP+SNNL achieves a notably higher F1-score of 79%, suggesting that SNNL significantly improves detection for WGAN-GP.

These results indicate that GAN-based augmentation methods, particularly when combined with SNNL, can preserve or even enhance majority-class performance. The inclusion of SNNL does not compromise the classification of well-represented classes, further supporting the overall robustness of the proposed approach in imbalanced intrusion detection tasks.

4.3. CSE-CIC-IDS2017 Results

4.3.1. Performance on Minority Class

Table 4 presents precision, recall, and F1-score comparisons for the minority classes (Bot and infiltration) using traditional oversampling methods (SMOTE, ADASYN, ROS) and several GAN variants (WGAN, CWGAN, WGAN-GP) as well as their SNNL-enhanced versions. The best results are highlighted in **bold**, and the second-best are underlined.

Table 4. Minority-class performance comparison on CSE-CIC-IDS2017 dataset.

Method	Bot			Infiltration		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score
SMOTE	0.333	**0.957**	0.473	0.723	**0.670**	0.630
ADASYN	0.347	**0.957**	0.480	0.723	**0.670**	0.630
ROS	0.440	0.923	0.560	0.693	0.640	**0.643**
WGAN	0.838	0.534	0.645	**0.833**	0.346	0.478
WGAN+SNNL	0.844	0.546	0.656	**0.833**	0.442	0.558
CWGAN	**0.863**	0.534	0.648	**0.833**	0.315	0.446
CWGAN+SNNL	0.857	0.560	**0.677**	**0.833**	0.363	0.487
WGAN-GP	0.848	0.545	0.658	**0.833**	0.363	0.493
WGAN-GP+SNNL	0.847	0.547	0.664	**0.833**	0.442	0.559

For the Bot class, traditional methods (SMOTE, ADASYN, ROS) achieve high recall but suffer from low precision, resulting in F1-scores below 0.56. In contrast, all GAN-based methods substantially improve precision, leading to consistently higher F1-scores. WGAN-GP+SNNL and CWGAN+SNNL attain the top performance with F1-scores of 0.664 and 0.677, respectively, indicating that the inclusion of SNNL enhances the quality of generated samples and promote a better precision–recall trade-off for minority-class detection.

For the infiltration class, traditional methods yield relatively balanced performance, with F1-scores in the range of 0.630–0.643. GAN-based methods, while achieving higher precision (e.g., 0.833), typically suffer from lower recall, which initially leads to reduced F1-scores. However, the integration of SNNL consistently improves recall across both WGAN and CWGAN variants. For example, WGAN-GP+SNNL improves recall from 0.363 to 0.442, raising the F1-score from 0.493 to 0.559. Similarly, CWGAN+SNNL improves upon CWGAN's F1-score from 0.446 to 0.487. Although these results for infiltration still fall short of the best traditional baselines in absolute terms, they indicate that SNNL helps alleviate recall degradation and improves the overall balance of precision and recall in GAN-based augmentation.

4.3.2. Overall Accuracy and Macro-F1

Table 5 presents the overall classification accuracy and macro-F1 scores on the CSE-CIC-IDS2017 dataset. All methods achieve high overall accuracy ($\geq$99.5%), with minimal variation across traditional and GAN-based augmentation techniques. This confirms that the proposed enhancements do not compromise general classification performance.

Table 5. Overall accuracy and macro-F1 comparison on CSE-CIC-IDS2017 dataset.

Method	Accuracy (%)	Macro-F1 (%)
SMOTE	99.5	85.9
ADASYN	99.5	85.1
ROS	99.6	87.7
WGAN	99.7	86.6
WGAN+SNNL	99.7	87.9
CWGAN	99.7	86.1
CWGAN+SNNL	99.7	87.1
WGAN-GP	99.7	87.0
WGAN-GP+SNNL	99.7	88.0

Macro-F1 provides a more balanced assessment of performance under class imbalance by treating each class equally. Across all the configurations, integrating SNNL yields slight but consistent improvements in macro-F1 compared to the base models. For instance,

WGAN-GP+SNNL achieves the highest macro-F1 of 88.0%, up from 87.0% in WGAN-GP. These results suggest that SNNL improves class-wise balance without adversely affecting majority-class performance or overall accuracy.

4.3.3. Majority-Class Performance

Figure 2 compares the F1-scores for majority classes (Brute Force, DoS, Normal, PortScan, and Web Attack) across traditional oversampling techniques and various GAN-based augmentation methods, including SNNL-enhanced variants.

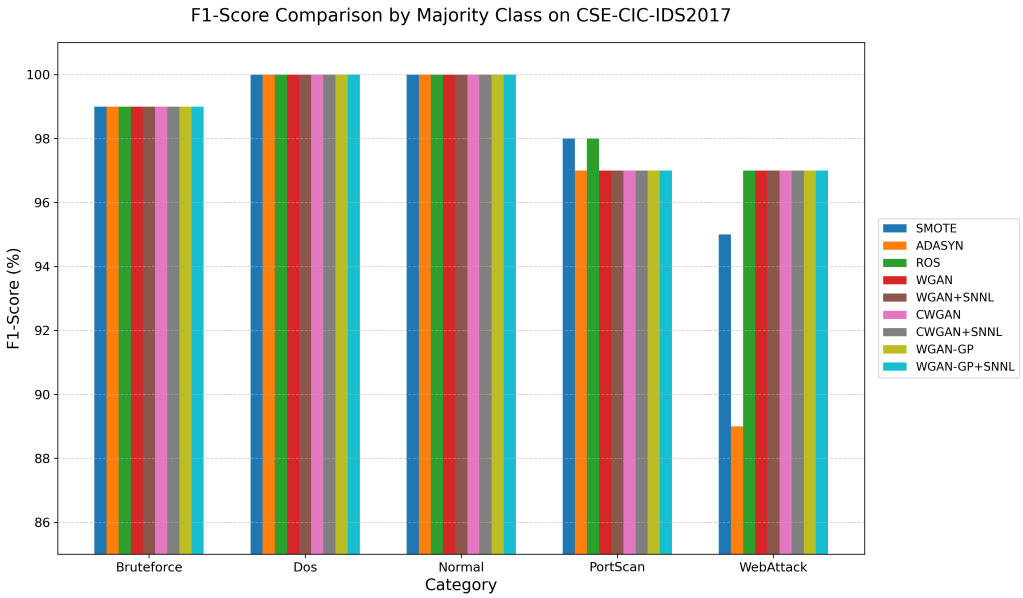


Figure 2. F1-score comparison by majority class on CSE-CIC-IDS2017 dataset.

For Brute Force, DoS, and Normal, all methods—both traditional and GAN-based—achieve near-perfect F1-scores (99–100%), indicating that these classes are reliably detected across augmentation strategies.

In the PortScan class, most methods achieve F1-scores around 97–98%, with no obvious differences observed between traditional and GAN-based methods. However, in the Web Attack category, ADASYN shows the lowest performance at 89%, while all other methods—including ROS and all GAN variants—achieve consistently higher F1-scores (95–97%). This suggests that GAN-based augmentation provides more stable performance on this class.

Overall, these results confirm that GAN-based methods, including those enhanced with SNNL, maintain competitive and stable detection for majority classes. The inclusion of SNNL does not degrade performance, further supporting the robustness and practicality of the proposed augmentation framework in class-imbalanced intrusion detection.

4.4. CSE-CIC-IDS2018 Results

4.4.1. Performance on Minority Class

Table 6 presents precision, recall, and F1-score comparisons for the minority classes (infiltration and Web Attack) across traditional oversampling methods and various GAN variants, including SNNL-enhanced versions. The best values are highlighted in **bold**, and the second-best are underlined.

Table 6. Minority-class performance comparison on CSE-CIC-IDS2018 dataset.

Method	Infiltration			Web Attack		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score
SMOTE	0.063	0.547	0.113	0.163	**0.810**	0.263
ADASYN	0.060	**0.557**	0.110	0.133	0.797	0.220
ROS	0.087	0.480	**0.137**	0.273	0.737	0.357
WGAN	0.362	0.075	0.113	0.912	0.583	0.692
WGAN+SNNL	0.363	0.077	0.115	0.933	0.656	0.765
CWGAN	0.357	0.072	0.110	0.940	0.543	0.653
CWGAN+SNNL	0.356	0.073	0.110	0.935	0.628	0.743
WGAN-GP	**0.52**	0.05	0.087	**0.983**	0.564	0.705
WGAM-GP+SNNL	0.493	0.057	0.100	0.953	0.74	**0.833**

For the infiltration class, traditional methods achieve relatively high recall (0.480–0.557) but suffer from low precision ($\leq$0.087), leading to limited F1-scores (0.110–0.137). GAN-based methods notably improve precision, particularly WGAN-GP and WGAN-GP+SNNL (above 0.49), although recall remains low ($\leq$0.057), resulting in comparable F1-scores to the traditional baselines. Overall, none of the evaluated approaches yield clear gains on this class, and SNNL integration appears to have limited impact.

By contrast, for Web Attack, GAN-based methods generally achieve higher F1-scores than traditional techniques. While ROS reaches 0.357, all GAN variants exceed 0.65. Among them, WGAN-GP+SNNL achieves the highest F1-score (0.833), attributed to improved recall (0.74) while maintaining high precision (0.953). This demonstrates that SNNL enhances generation quality and supports better minority-class separability for detecting Web Attack.

4.4.2. Overall Accuracy and Macro-F1

Table 7 presents the overall classification accuracy and macro-F1 scores on the CSE-CIC-IDS2018 dataset. All methods achieve consistently high accuracy ($\geq$98.7%), with minimal variation across both traditional and GAN-based augmentation strategies. The inclusion of SNNL does not adversely affect overall accuracy.

Table 7. Overall accuracy and macro-F1 comparison on CSE-CIC-IDS2018 dataset.

Method	Accuracy (%)	Macro-F1 (%)
SMOTE	98.7	75.9
ADASYN	98.7	75.0
ROS	97.7	77.6
WGAN	98.7	82.8
WGAN+SNNL	98.7	83.8
CWGAN	98.7	82.2
CWGAN+SNNL	98.7	83.5
WGAN-GP	98.8	82.6
WGAN-GP+SNNL	98.8	84.6

Macro-F1 offers a more balanced perspective by averaging performance across all classes equally, making it particularly useful under imbalanced distributions. As shown in Table 7, models enhanced with SNNL tend to outperform their baseline counterparts in macro-F1. Notably, WGAN-GP+SNNL achieves the highest macro-F1 score of 84.6%, followed by WGAN+SNNL and CWGAN+SNNL. These results show that SNNL contributes to improved minority-class detection without sacrificing majority-class accuracy or overall performance.

4.4.3. Majority-Class Performance

Figure 3 presents the F1-score comparison for majority classes (benign, Bot, Brute Force, DDoS, and DoS) across traditional oversampling techniques and GAN-based augmentation strategies.

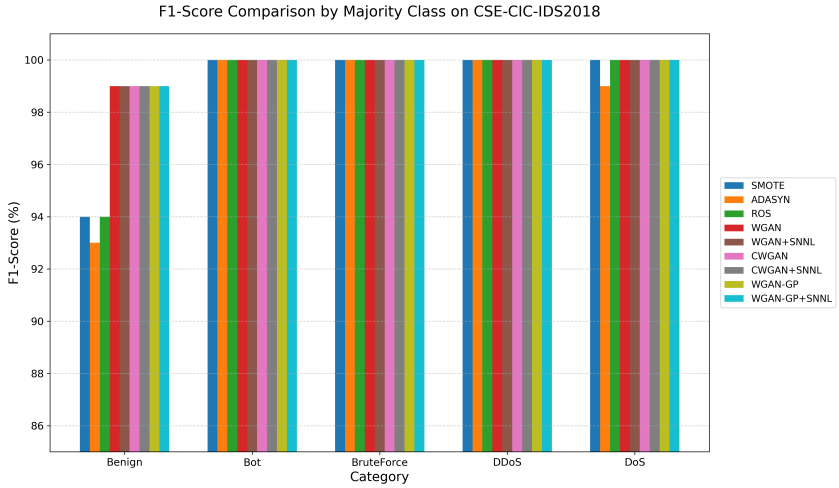


Figure 3. F1-score comparison by majority class on CSE-CIC-IDS2018 dataset.

Across Bot, Brute Force, DDoS, and DoS, all methods—including both traditional oversampling techniques and GAN-based models—achieve perfect or near-perfect F1-scores (99–100%), indicating that these attack categories are well captured by the classifier regardless of augmentation strategy.

In the benign class, however, noticeable differences emerge. Traditional methods such as SMOTE, ADASYN, and ROS result in relatively lower F1-scores (93–94%), whereas all GAN-based methods—including SNNL-enhanced variants—consistently achieve higher F1-scores around 99%. This suggests that GAN-generated samples provide more representative decision boundaries even for complex majority classes such as benign, where subtle noise and variability are common.

Overall, these results indicate that GAN-based augmentation methods offer stable and high-performing detection across majority classes. The inclusion of SNNL does not adversely affect majority-class performance, reinforcing the reliability and consistency of the proposed approach in realistic NIDS settings.

4.5. Computational Overhead

To evaluate the computational overhead introduced by SNNL, we compare the baseline WGAN and its SNNL-enhanced version on the CSE-CIC-IDS2017 dataset using identical experimental settings. Each configuration was run three times, and the reported values reflect the average across runs. Table 8 summarizes the results.

Table 8. Average computational overhead of WGAN vs. WGAN+SNNL on CSE-CIC-IDS2017.

Metric	WGAN	WGAN+SNNL	Overhead
Avg. Training Time per Epoch (s)	102.26	128.41	+25.5%
Peak GPU Memory Usage (MB)	1822.77	1823.23	≈0%

These results show that integrating SNNL introduces a moderate computational cost: the average training time per epoch increases by approximately 25.5%. In contrast, GPU memory usage remains nearly identical, suggesting that the added cost is computational rather than memory-bound.

Given the improved minority-class detection observed in prior sections, this overhead is acceptable in practice—particularly in offline training scenarios where detection performance is prioritized over minimal runtime.

4.6. Discussion of Experimental Results

The experimental results on NSL-KDD, CSE-CIC-IDS2017, and CSE-CIC-IDS2018 demonstrate the advantages of GAN-based data augmentation methods over traditional oversampling techniques, particularly in minority-class detection. While some improvements in F1-score may appear numerically small (e.g., from 0.168 to 0.200), these gains are directionally consistent across datasets and GAN variants and frequently observed in the most underrepresented categories. In practical intrusion detection scenarios, even modest performance gains on rare attack types can significantly reduce the risk of undetected threats.

The integration of SNNL further improves minority-class performance by enhancing the feature-space alignment between synthetic and real samples. The consistent upward trends across multiple metrics, including macro-F1, indicate that these improvements are not incidental. For instance, WGAN-GP+SNNL achieves macro-F1 gains of 3.3%, 1.0%, and 2.0% over WGAN-GP on NSL-KDD, CSE-CIC-IDS2017, and CSE-CIC-IDS2018, respectively. Notably, in cases where recall differences appear small for a single class (e.g., 0.110 vs. 0.112), performance remains at a comparable level while other categories exhibit clearer improvements—together contributing to a more balanced and reliable classifier overall.

In the infiltration class of CSE-CIC-IDS2017, traditional methods exhibit slightly higher recall, leading to marginally better F1-scores. However, SNNL-integrated GANs consistently outperform their non-SNNL counterparts on this class, suggesting that even in challenging scenarios, SNNL contributes to performance improvement.

In terms of computational cost, we conducted three repeated experiments comparing WGAN and WGAN+SNNL on the CSE-CIC-IDS2017 dataset. On average, incorporating SNNL increased training time per epoch from 102.26 s to 128.41 s (a 25.5% overhead), while peak GPU memory usage remained effectively unchanged. These results show that the additional cost is moderate and manageable, particularly in offline training scenarios where detection effectiveness is prioritized over latency.

Lastly, performance on majority classes remains stable—or even slightly improved—after incorporating SNNL. This confirms that the proposed method improves class-wise detection balance without sacrificing overall accuracy or majority-class performance, underscoring its practicality for real-world NIDS applications with severe class imbalance.

5. Visualization

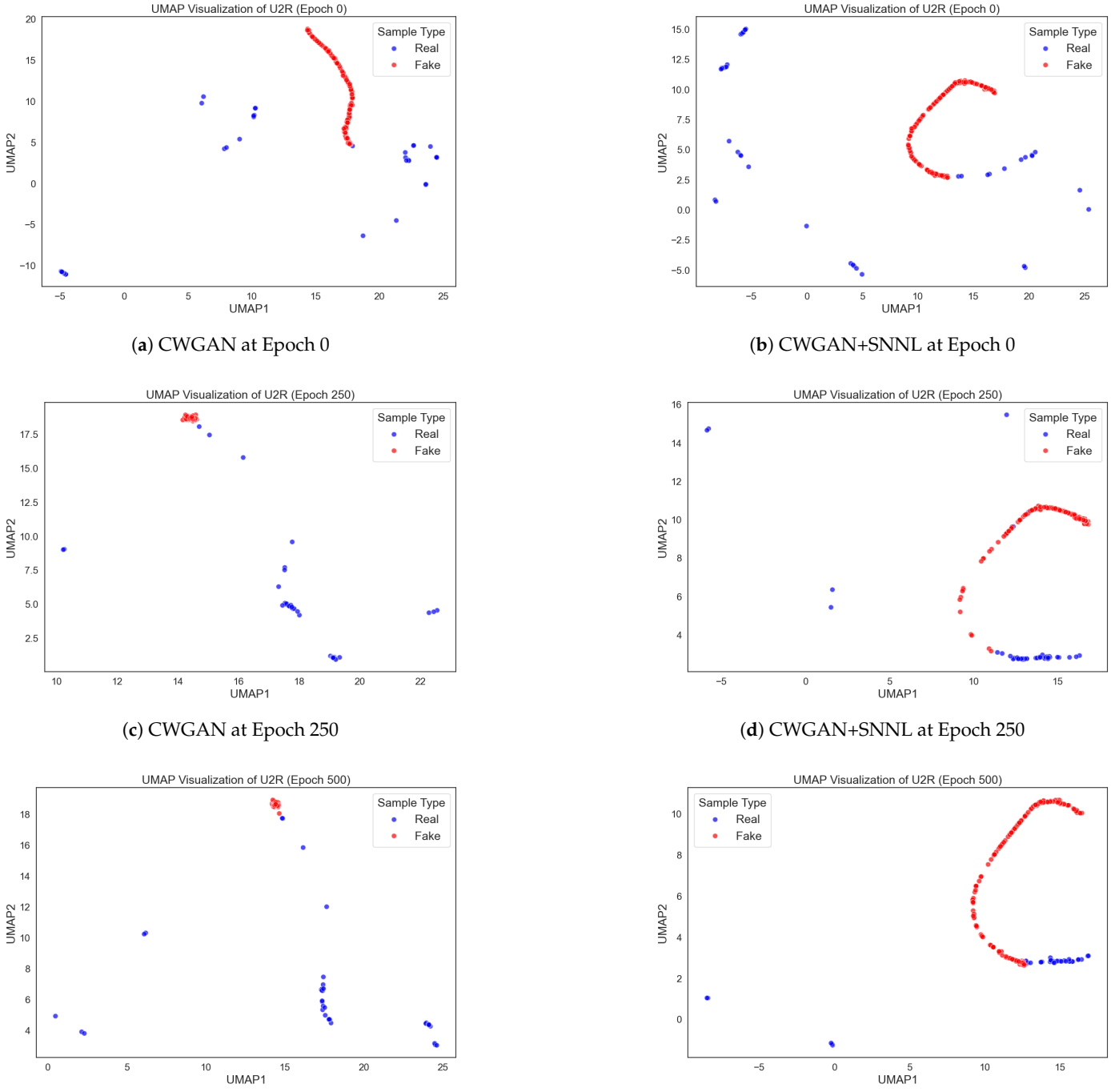
Prior work has demonstrated that visualizing learned feature representations can provide valuable insight into network intrusion behaviours and the quality of synthetic data [29]. To analyse the effectiveness of SNNL, we visualize the feature distribution of the User-to-Root (U2R) class in the NSL-KDD dataset. We compare the feature distributions of samples generated by CWGAN and CWGAN+SNNL at different training epochs, using Uniform Manifold Approximation and Projection (UMAP) for dimensionality reduction.

5.1. Experimental Setup

For both CWGAN and CWGAN+SNNL, we generate 1000 synthetic U2R samples at three different epochs: 0, 250, and 500. From each generated set, we randomly select 200 fake samples, mix them with real U2R samples, and use UMAP to project them into a 2D feature space. This visualization allows us to assess how well the generated data aligns with real data as training progresses.

5.2. Results and Analysis

Figure 4 illustrates a detailed comparison of UMAP feature distributions for synthetic U2R samples generated by CWGAN and CWGAN+SNNL across various epochs (Epoch 0, Epoch 250, Epoch 500).



(**a**) CWGAN at Epoch 0

(**b**) CWGAN+SNNL at Epoch 0

(**c**) CWGAN at Epoch 250

(**d**) CWGAN+SNNL at Epoch 250

(**e**) CWGAN at Epoch 500

(**f**) CWGAN+SNNL at Epoch 500

Figure 4. UMAP feature visualization of U2R samples generated by CWGAN (**left**) and CWGAN+SNNL (**right**) at different epochs: (**a**,**b**) Epoch 0, (**c**,**d**) Epoch 250, and (**e**,**f**) Epoch 500.

At Epoch 0, samples generated by both CWGAN and CWGAN+SNNL display highly dispersed feature distributions, significantly deviating from the true U2R sample space. At Epoch 250, samples generated by CWGAN (Figure 4c) form a cluster that is distinctly separate from the genuine U2R data. CWGAN+SNNL samples (Figure 4d) clearly start

overlapping with real data, indicating that SNNL encourages the correlation between generated and genuine data.

At Epoch 500, CWGAN samples (Figure 4e) still demonstrate noticeable separation from real U2R samples, reflecting limitations in fully capturing complex minority-class distributions. Conversely, CWGAN+SNNL-generated samples (Figure 4f) exhibit significant alignment with real data, highlighting the effectiveness of SNNL in guiding GAN training toward generating highly realistic minority-class samples.

Overall, the visualization clearly shows that SNNL functions as a powerful feature-space regularizer. Including SNNL into CWGAN enhances the similarity and consistency of feature representations.

6. Limitation

While the proposed SNNL-enhanced GAN framework demonstrates consistent improvements in minority-class detection, several limitations remain.

First, the method assumes access to labelled samples from both majority and minority classes during training. This limits its applicability in unsupervised or purely anomaly-based intrusion detection settings, where only benign data is typically available. Extending the framework to semi-supervised or unsupervised scenarios remains an open research direction.

Second, incorporating SNNL introduces moderate computational overhead. Although the increase in GPU memory usage is negligible, training time per epoch increases due to the additional pairwise similarity computations. While this overhead is acceptable for offline training, it may hinder deployment in latency-sensitive or resource-constrained environments.

Finally, the effectiveness of SNNL-enhanced augmentation still depends on the generator's capacity to model complex minority-class distributions. In cases of extreme imbalance or data noise, GAN training may become unstable or suffer from mode collapse, limiting the benefits of SNNL. Further work is needed to improve robustness under such conditions.

7. Conclusions and Future Work

This study presents SNNL as an enhancement to GAN-based data augmentation for addressing class imbalance in network intrusion detection. By integrating SNNL into WGAN, CWGAN, and WGAN-GP frameworks, the proposed method improves feature-space alignment between real and synthetic samples, thereby enhancing the classifier's ability to detect minority attack classes without compromising overall or majority-class performance.

Comprehensive experiments on three benchmark datasets, NSL-KDD, CSE-CIC-IDS2017, and CSE-CIC-IDS2018, demonstrate that GAN-based augmentation consistently outperforms traditional oversampling techniques in most settings. The addition of SNNL provides further improvements in minority-class F1-scores across various attack types. While a few cases such as the infiltration class show marginal advantage for traditional methods, SNNL-enhanced GANs generally achieve a better balance between precision and recall, highlighting their practical benefit in imbalanced scenarios.

Although SNNL introduces moderate computational overhead, our empirical analysis indicates that the increase in training time remains acceptable for offline training pipelines. Future work will explore extending the framework to semi-supervised and unsupervised intrusion detection settings, where labelled minority-class data is limited or unavailable. Additionally, we aim to reduce training cost through scalable similarity approximation techniques. Beyond NIDS, the proposed SNNL-enhanced GAN framework may also generalize to other imbalanced learning domains such as fraud detection, rare disease diagnosis, and industrial anomaly detection.

Author Contributions: Conceptualization, J.L.; methodology, J.L.; software, J.L.; formal analysis, J.L.; visualization, J.L.; writing—original draft preparation, J.L.; writing—review and editing, W.Z. and Y.-W.C.; supervision, W.Z., Y.-W.C. and W.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The datasets used in this study are publicly available and have been cited appropriately in the References section. No new data were generated or collected during this study.

Acknowledgments: During the preparation of this manuscript, the authors used ChatGPT (OpenAI, GPT-4o, April 2025 version) to assist with language editing and structural organization. The authors have thoroughly reviewed and edited the AI-generated content and take full responsibility for the accuracy and integrity of the final manuscript.

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

References

1. Zong, W.; Chow, Y.-W.; Susilo, W. A 3D Approach for the Visualization of Network Intrusion Detection Data. In Proceedings of the 2018 International Conference on Cyberworlds (CW), Singapore, 3–5 October 2018; IEEE: New York, NY, USA, 2018; pp. 308–315.
2. Zong, W.; Chow, Y.-W.; Susilo, W. A Two-Stage Classifier Approach for Network Intrusion Detection. In *Information Security Practice and Experience, Proceedings of the 14th International Conference, ISPEC 2018, Tokyo, Japan, 25–27 September 2018*; Su, C., Kikuchi, H., Eds.; Springer: Cham, Switzerland, 2018; Volume 11125, pp. 197–213. [CrossRef]
3. Lee, J.; Park, K. GAN-Based Imbalanced Data Intrusion Detection System. *Pers. Ubiquitous Comput.* **2021**, *25*, 121–128. [CrossRef]
4. Ding, S.; Kou, L.; Wu, T. A GAN-Based Intrusion Detection Model for 5G Enabled Future Metaverse. *Mob. Netw. Appl.* **2022**, *27*, 2596–2610. [CrossRef]
5. Liao, D.; Huang, S.; Tan, Y.; Bai, G. Network Intrusion Detection Method Based on GAN Model. In Proceedings of the 2020 International Conference on Computer Communication and Network Security (CCNS), Xi'an, China, 21–23 August 2020; pp. 153–156. [CrossRef]
6. de Araujo-Filho, P.F.; Naili, M.; Kaddoum, G.; Fapi, E.T.; Zhu, Z. Unsupervised GAN-Based Intrusion Detection System Using Temporal Convolutional Networks and Self-Attention. *IEEE Trans. Netw. Serv. Manag.* **2023**, *20*, 4951–4963. [CrossRef]
7. Liu, X.; Li, T.; Zhang, R.; Wu, D.; Liu, Y.; Yang, Z.; Sciancalepore, S. A GAN and Feature Selection-Based Oversampling Technique for Intrusion Detection. *Secur. Commun. Netw.* **2021**, *2021*, 9947059. [CrossRef]
8. Li, S.; Cao, Y.; Liu, S.; Lai, Y.; Zhu, Y.; Ahmad, N. HDA-IDS: A Hybrid DoS Attacks Intrusion Detection System for IoT by Using Semi-Supervised CL-GAN. *Expert Syst. Appl.* **2024**, *238*, 122198. [CrossRef]
9. Poongodi, M.; Hamdi, M. Intrusion Detection System Using Distributed Multilevel Discriminator in GAN for IoT System. *Trans. Emerg. Telecommun. Technol.* **2023**, *34*, e4815. [CrossRef]
10. Rahman, S.; Pal, S.; Mittal, S.; Chawla, T.; Karmakar, C. SYN-GAN: A Robust Intrusion Detection System Using GAN-Based Synthetic Data for IoT Security. *Internet Things* **2024**, *26*, 101212. [CrossRef]
11. Li, S.; Li, Q.; Li, M. A Method for Network Intrusion Detection Based on GAN-CNN-BiLSTM. *Int. J. Adv. Comput. Sci. Appl.* **2023**, *14*, 507–515. [CrossRef]
12. Sophia, N.A.; Sivasai, A.S.V.; Sachin, S. Intrusion Detection System Using Generative Adversarial Network (GAN). In Proceedings of the 2024 2nd International Conference on Advancement in Computation & Computer Technologies (InCACCT), Gharuan, India, 2–3 May 2024; pp. 917–922. [CrossRef]
13. Chen, H.; Jiang, L. Efficient GAN-Based Method for Cyber-Intrusion Detection. *arXiv* **2019**, arXiv:1904.02426. [CrossRef]
14. Xu, W.; Jang-Jaccard, J.; Liu, T.; Sabrina, F.; Kwak, J. Improved Bidirectional GAN-Based Approach for Network Intrusion Detection Using One-Class Classifier. *Computers* **2022**, *11*, 85. [CrossRef]
15. Ding, H.; Sun, Y.; Huang, N.; Shen, Z.; Cui, X. TMG-GAN: Generative Adversarial Networks-Based Imbalanced Learning for Network Intrusion Detection. *IEEE Trans. Inf. Forensics Secur.* **2024**, *19*, 1156–1167. [CrossRef]
16. Boppana, T.K.; Bagade, P. GAN-AE: An Unsupervised Intrusion Detection System for MQTT Networks. *Eng. Appl. Artif. Intell.* **2023**, *119*, 105805. [CrossRef]
17. Peng, Y.; Fu, G.; Luo, Y.; Hu, J.; Li, B.; Yan, Q.; Wenzheng, L. Detecting Adversarial Examples for Network Intrusion Detection System with GAN. In Proceedings of the 2020 IEEE 11th International Conference on Software Engineering and Service Science (ICSESS), Beijing, China, 16–18 October 2020; pp. 6–10. [CrossRef]

18. Ahmad, R.; Li, L.H.; Sharma, A.K.; Tanone, R. Boundary-Seeking GAN Approach to Improve Classification of Intrusion Detection Systems Based on Machine Learning Model. In Proceedings of the 2023 17th International Conference on Ubiquitous Information Management and Communication (IMCOM), Seoul, Republic of Korea, 3–5 January 2023; pp. 1–5. [CrossRef]
19. Feng, J.; Wang, C.; Xue, H.; Zhang, L. Efficient Anomaly Intrusion Detection Using Transformer-Based GAN Network. In Proceedings of the 2024 IEEE 7th International Electrical and Energy Conference (CIEEC), Harbin, China, 10–12 May 2024; pp. 3876–3881. [CrossRef]
20. Bai, W.; Wang, K.; Chen, K.; Li, S.; Li, B.; Zhang, N. SC-WGAN: GAN-Based Oversampling Method for Network Intrusion Detection. In *Engineering of Complex Computer Systems*; Bai, G., Ishikawa, F., Ait-Ameur, Y., Papadopoulos, G.A., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2025; Volume 14784. [CrossRef]
21. Zhang, X.; Chen, X.; He, Y.; Wang, Y.; Cai, Y.; Li, H.; Chen, X.; Srivastava, H.M. GAN-Based Intrusion Detection Model Using MLP Encoder. In *Proceedings of SPIE—The International Society for Optical Engineering*; SPIE: Bellingham, WA, USA, 2023; Volume 12756, p. 127564T. [CrossRef]
22. Strickland, C.; Zakar, M.; Saha, C.; Soltani Nejad, S.; Tasnim, N.; Lizotte, D.J.; Haque, A. DRL-GAN: A Hybrid Approach for Binary and Multiclass Network Intrusion Detection. *Sensors* **2024**, *24*, 2746. [CrossRef] [PubMed]
23. Frosst, N.; Papernot, N.; Hinton, G. Analyzing and Improving Representations with the Soft Nearest Neighbor Loss. *arXiv* **2019**, arXiv:1902.01889. [CrossRef]
24. Schroff, F.; Kalenichenko, D.; Philbin, J. FaceNet: A Unified Embedding for Face Recognition and Clustering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; IEEE: New York, NY, USA, 2015.
25. Wen, Y.; Zhang, K.; Li, Z.; Qiao, Y. A Discriminative Feature Learning Approach for Deep Face Recognition. In *Computer Vision–ECCV 2016*; Leibe, B., Matas, J., Sebe, N., Welling, M., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2016; Volume 9911. [CrossRef]
26. Hermans, A.; Beyer, L.; Leibe, B. In Defense of the Triplet Loss for Person Re-Identification. *arXiv* **2017**, arXiv:1703.07737. [CrossRef]
27. Tavallaee, M.; Bagheri, E.; Lu, W.; Ghorbani, A. A Detailed Analysis of the KDD CUP 99 Data Set. In Proceedings of the 2nd IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA), Ottawa, ON, Canada, 8–10 July 2009.
28. Sharafaldin, I.; Lashkari, A.H.; Ghorbani, A.A. 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)*; SCITEPRESS: Setúbal, Portugal, 2018.
29. Zong, W.; Chow, Y.-W.; Susilo, W. Interactive Three-Dimensional Visualization of Network Intrusion Detection Data for Machine Learning. *Future Gener. Comput. Syst.* **2020**, *102*, 292–306. [CrossRef]