Computer Modeling in Engineering & Sciences

Tech Science Press

Doi:10.32604/cmes.2025.064874

ARTICLE

A Hybrid Wasserstein GAN and Autoencoder Model for Robust Intrusion Detection in IoT

Mohammed S. Alshehri[1,*], Oumaima Saidani[2], Wajdan Al Malwi[3], Fatima Asiri[3], Shahid Latif[4], Aizaz Ahmad Khattak[5] and Jawad Ahmad[6]

[1]Department of Computer Science, College of Computer Science and Information Systems, Najran University, Najran, 61441, Saudi Arabia

[2]Department of Information Systems, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P. O. Box 84428, Riyadh, 11671, Saudi Arabia

[3]Department of Informatics and Computer Systems, College of Computer Science, King Khalid University, Abha, 62521, Saudi Arabia

[4]School of Computing and Creative Technologies, University of the West of England, Bristol, BS16 1QY, UK

[5]School of Computing, Engineering & The Built Environment, Edinburgh Napier University, 10 Colinton Road, Edinburgh, EH10 5DT, UK

[6]Cybersecurity Center, Prince Mohammad Bin Fahd University, Al Khobar, 31952, Saudi Arabia

*Corresponding Author: Mohammed S. Alshehri. Email: msalshehry@nu.edu.sa

Received: 26 February 2025; Accepted: 23 April 2025; Published: 30 June 2025

ABSTRACT: The emergence of Generative Adversarial Network (GAN) techniques has garnered significant attention from the research community for the development of Intrusion Detection Systems (IDS). However, conventional GAN-based IDS models face several challenges, including training instability, high computational costs, and system failures. To address these limitations, we propose a Hybrid Wasserstein GAN and Autoencoder Model (WGAN-AE) for intrusion detection. The proposed framework leverages the stability of WGAN and the feature extraction capabilities of the Autoencoder Model. The model was trained and evaluated using two recent benchmark datasets, 5GNIDD and IDSIoT2024. When trained on the 5GNIDD dataset, the model achieved an average area under the precision-recall curve is 99.8% using five-fold cross-validation and demonstrated a high detection accuracy of 97.35% when tested on independent test data. Additionally, the model is well-suited for deployment on resource-limited Internet-of-Things (IoT) devices due to its ability to detect attacks within microseconds and its small memory footprint of 60.24 kB. Similarly, when trained on the IDSIoT2024 dataset, the model achieved an average PR-AUC of 94.09% and an attack detection accuracy of 97.35% on independent test data, with a memory requirement of 61.84 kB. Extensive simulation results demonstrate that the proposed hybrid model effectively addresses the shortcomings of traditional GAN-based IDS approaches in terms of detection accuracy, computational efficiency, and applicability to real-world IoT environments.

KEYWORDS: Autoencoder; cybersecurity; generative adversarial network; Internet of Things; intrusion detection system

1 Introduction

Intrusion detection systems (IDSs) play a very important role in protecting IoT networks from a number of cyberattacks that are mainly targeted at device and network vulnerabilities such as the use of insecure communication protocols or weak authentication mechanisms [1]. Most of these attacks are launched through malicious software or firmware updates that may result in unauthorized access to the

network or control of IoT devices [2]. Cyberattacks can violate the confidentiality, integrity, and availability of the network and may result in data breaches, unauthorized access to sensitive information, and denial of service [3,4]. The main challenge in this regard is the resource-constrained nature of IoT devices that pose a challenge to implementing advanced and robust security frameworks. The increasing complexity of security algorithms results in system outages, reduced performance, or poor service quality in IoT networks [5].

Over the past few years, GANs have received great attention because of their ability to learn data distribution and generate synthetic data that can mimic possible attack patterns [6]. Traditional signature-based approaches are also ineffective in identifying new and unexpected cyberattacks, whereas GANs learn to generate adversarial examples to identify outliers [7]. However, there are several shortcomings of the traditional GAN variants as well. For example, vanilla GAN [8] has some drawbacks such as model collapse and training instability that limit its effectiveness in generating diverse and representative samples. Conditional GANs [9] are more context-sensitive than their counterparts but they need labeled data. CycleGANs [10] are very efficient in the domain translation task, even in the absence of paired data, but they are usually slow and not very accurate in identifying small anomalies. The Wasserstein GAN [11] also improves the stability and diversity but at the expense of increasing the model complexity.

To overcome these issues, this paper proposes a Wasserstein GAN with Autoencoders (WGAN-AE) that integrates the best aspects of WGANs and autoencoders to develop a more efficient and effective intrusion detection system for IoT networks. The WGAN-AE can produce stable and diverse samples by leveraging the Wasserstein distance, which measures the difference between two probability distributions. Unlike traditional distance metrics, it provides a smoother and more meaningful way to compare real and generated data, leading to better training stability and improved detection of various attack types [12]. Autoencoders assist in detecting the changes that are likely to be a sign of a cyberattack by learning the detailed patterns of the normal IoT traffic [13]. This approach is less dependent on large datasets suitable for dynamic IoT environments. Furthermore, the WGAN-AE offers a complete solution that entails the reconstruction and anomaly identification of any input by comparing it with the learned normal behavior. This makes training more stable and enhances the novel attack detection rate that may not be detected by other models. A comparison of the proposed WGAN-AE with state-of-the-art GAN variants is presented in Table 1. The major contributions of the article are summarized as follows.

1. A hybrid intrusion detection framework, WGAN-AE, is proposed by utilizing the key strengths of WGAN and AE. The model enhances the detection of emerging attack vectors by using Wasserstein distance for more stable training and autoencoder-based feature extraction.
2. The research introduces an unsupervised GAN-based approach that learns the normal operation of the system and identifies changes as possible cyberattacks. It also achieves this by properly distinguishing between the normal and attack behaviors through the use of adversarial training of the generator to generate diverse attack types.
3. A comprehensive evaluation framework is developed to assess the computational efficiency and effectiveness of the proposed scheme by using two advanced and comprehensive IDS benchmark datasets.

The remainder of the article is organized as follows. In Section 2, we present some latest contributions related to GAN-based IDS frameworks and describe some preliminaries. In Section 3, we elaborate on the research methodology and the design of the proposed framework. In Section 4, we present a brief discussion of the experimental setup and outcomes. Finally, in Section 5, we conclude the research with future research directions.

Table 1: A comparison of state-of-the-art GAN variants with the proposed WGAN-AE in the context of IDS

Feature	GAN variants				
	Vanilla GAN	**Conditional GAN**	**Cycle GAN**	**Wasserstein GAN**	**WGAN-AE**
Training stability	Low	Moderate	Moderate	High	Very high
Mode collapse	High	Moderate	Moderate	Low	Very low
Feature learning	Limited	Good	Good	Strong	Strong
Anomaly detection	Poor	Moderate	Good	Good	Excellent
Training complexity	Moderate	High	High	Moderate	Moderate
Robustness	Low	Moderate	Moderate	High	Very high

2 Related Work

This section overviews some significant contributions related to GAN-based IDS architectures. Rahman et al. [6] explored the potential of GAN for intrusion detection in IoT. The proposed scheme significantly decreased the dependency on real-world data. To analyze the effectiveness of designed model, the authors conducted extensive experiments on three open-source datasets including UNSW-NB15, NSL-KDD and BoT-IoT datasets. Message Queuing Telemetry Transport (MQTT) is a widely adapted network protocol in IoT infrastructures because of its lightweight and flexible nature. Boppana and Bagade [14] proposed a novel unsupervised GAN and autoencoder-based model GAN-AE, to detect unknown intrusions in MQTT-based IoT applications. The experimental results demonstrate the effectiveness of the proposed method against various modern IDS approaches. In another study, Li et al. [15] proposed a hybrid IDS model to detect Denial of Service (DoS)/botnet attacks in IoT systems. The authors designed an anomaly-based detection model called CL-GAN (CNN-LSTM GNN), combining a Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) with GAN to define a baseline of normal activity and identify malicious traffic. The proposed architecture was evaluated with NSL-KDD, CICIDS2018, and Bot-IoT datasets.

de Araujo-Filho et al. [16] presented a novel fog-based unsupervised IDS using GANs. The proposed IDS was developed for a fog architecture to meet the low-latency requirements of cyberphysical systems. Experimental outcomes indicated the higher detection rates and superior performance of the proposed approach over baseline models using three datasets. Zeghida et al. [17] proposed GAN-based methods to achieve a balanced dataset for greater attack detection accuracy. Additionally, they introduced three dedicated IDSs for attack detection using the MQTT protocol based on hybrid deep learning (DL) algorithms: Convolutional Neural Network with Recurrent Neural Network (CNN-RNN), CNN with Long Short-Term Memory (CNN-LSTM), and CNN with Gated Recurrent Unit (CNN-GRU). The experimental results demonstrated that the generated dataset had a superior performance in multiclass configuration. In another study, Das et al. [18] proposed two models: a Feedforward Neural Network (FNN) network and a CNN. The proposed models have been trained and tested on standard datasets as well as synthetic datasets. The generation of this synthetic data employs a Conditional Tabular Generative Adversarial Network (CTGAN). The experimental outcomes indicated less training time and memory utilization than several baseline models.

Wang et al. [19] proposed a Multi-Critics GAN to address the data imbalance issues in IDS systems. The authors analyzed the generated data quality by using Principal Component Analysis (PCA) plots and correlation heatmaps. Subsequently, they incorporated a hybrid CNN-LSTM model to analyze the clusters to achieve the promising performance of IDS systems. The experimental results confirmed the higher attack detection rate with better generalization. Dong et al. [20] presented a novel IDS framework MasqueradeGAN-GP (Generative Adversarial Networks with Gradient Penalty), for 6G networks

by integrating a WGAN with a Gradient Penalty. In the proposed architecture, a generator transforms the anomalous traffic into a semblance of benign activity and the discriminator discerns the genuine and adversarial traffic. The efficacy of the designed models was investigated by conducting extensive experiments using two open-source datasets. Brabin et al. [21] presented a Cycle-Consistent GAN-based attack detection and secure data transmission framework for smart cities. The designed framework incorporated a Wild horse optimizer for feature extraction. Subsequently, the selected features are provided to the cycle-consistent GAN classifier to distinguish normal and malicious traffic. Furthermore, to ensure a secure data transmission, the authors incorporated Advanced Encryption Standard (AES) with Chameleon Swarm Algorithm.

The aforementioned works present a significant contribution towards GAN-based IDSs. However, the existing studies have a few shortcomings, including reliance on outdated datasets that fail to capture real-time IoT security challenges and a primary focus on attack detection accuracy while neglecting critical constraints such as memory requirements and computational efficiency. While some approaches address data imbalance using advanced GAN variants like Multi-Critics GAN, CTGAN, and WGAN with Gradient Penalty, others integrate federated learning for enhanced security in Fog-assisted IoT networks. However, these studies lack a holistic evaluation framework. To overcome these shortcomings, this article proposes a WGAN-AE, which is trained and evaluated using two real-time benchmark datasets. The proposed approach ensures a comprehensive performance analysis, including accuracy, memory consumption, system latency, and cross-validation, making it a more robust solution for modern IoT-based IDS.

3 Research Methodology and the Proposed Framework

This section briefly describes the dataset description, the design of the proposed architecture, and the training process. The workflow of the proposed architecture is presented in Fig. 1.

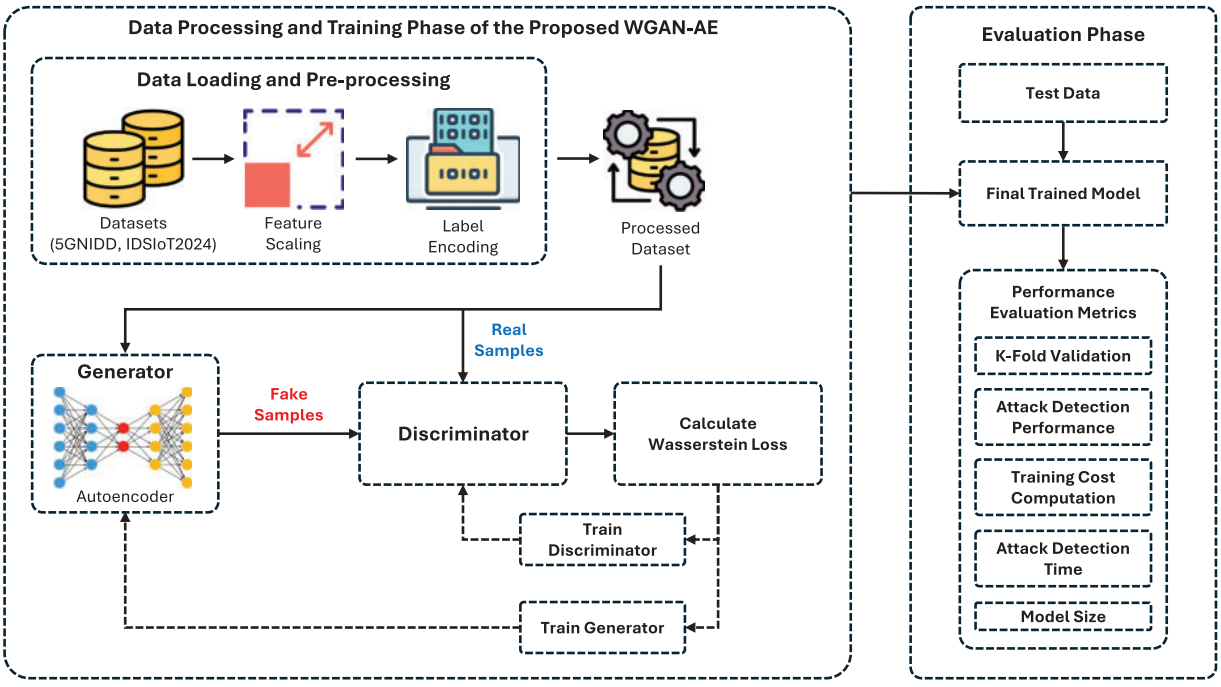

Figure 1: Block diagram of the proposed architecture

3.1 Dataset Description

To train and evaluate the proposed architecture's performance, we utilized three of the latest and most comprehensive datasets. The following provides a detailed description.

3.1.1 5G-NIDD Dataset

The 5G-NIDD dataset is a fully labeled collection of network traffic data generated from a functional 5G test network at the University of Oulu, Finland [22]. The dataset includes various attack scenarios, such as Denial of Service (DoS) attacks, Internet Control Message Protocol (ICMP) Flood, synchronize (SYN) Flood, User Datagram Protocol (UDP) Flood, Hypertext Transfer Protocol (HTTP) Flood, and Slowrate and port scans, including Transmission Control Protocol (TCP) Connect Scan, SYN Scan, and UDP Scan. The detailed distribution of the 5G-NIDD dataset is presented in Fig. 2.

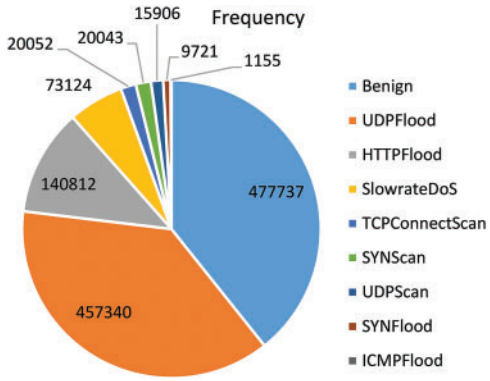

Figure 2: Distribution of 5G-NIDD dataset

3.1.2 IDSIoT2024 Dataset

The IDSIoT2024 dataset [23] is a comprehensive, real-time collection of network traffic data from an Internet of Things (IoT) environment comprising seven diverse smart devices: a smartwatch, surveillance camera, smartphone, laptop, smart vacuum, smart TV, and smart light. In this setup, the laptop serves a dual role: continuously monitoring and logging network traffic for analysis and actively executing various network-based attacks to simulate potential security threats. The dataset includes seven main categories: DoS, Injection, Man-in-the-Middle (MITM), malware, normal, routing, and vulnerability analysis. The detailed distribution of the IDSIoT2024 dataset is presented in Fig. 3.

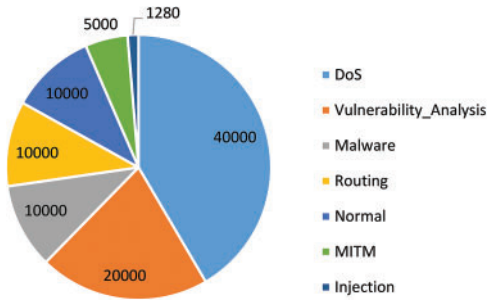

Figure 3: Distribution of IDSIoT2024 dataset

3.2 The Proposed Architecture

This section provides a detailed description of each module of the proposed architecture.

3.2.1 Data Preprocessing

The preprocessing of the dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ is an essential step to prepare both the input features$X$ and target labels y for the hybrid WGAN-AE and autoencoder model. This stage typically involves scaling the features and encoding the categorical target labels for classification tasks.

Feature Scaling (Standardization): The first step is to normalize the feature data so that each feature has zero mean and unit variance. This ensures that all features contribute equally during training, preventing features with larger ranges from dominating the optimization process.

Given a feature matrix $X \in \mathbb{R}^{N\times d}$, where N is the number of samples and d is the number of features, each feature column $x_j \in \mathbb{R}^N$ is standardized using z -score normalization. The transformation is performed as follows (1):

$$x_j^{\text{scaled}} = \frac{x_j - \mu_j}{\sigma_j}, \tag{1}$$

where:

- $\mu_j = \frac{1}{N}\sum_{i=1}^{N} x_{ij}$ is the mean of the j-th feature across all samples.
- $\sigma_j = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(x_{ij} - \mu_j\right)^2}$ is the standard deviation of the j-th feature.

This process ensures that the scaled feature x_j^{scaled} has (2):

$$\mathbb{E}\left[x_j^{\text{scaled}}\right] = 0 \quad \text{and} \quad \text{Var}\left(x_j^{\text{scaled}}\right) = 1. \tag{2}$$

The entire feature matrix X_{scaled} is then used as the input for the model.

Label Encoding: In classification tasks, the target labels are often categorical, so we need to convert the labels into a numerical form that can be used by machine learning models. Label encoding is a common method for transforming categorical labels into integers.

Let $y_i \in \{1, 2, \ldots, C\}$ be the categorical class label of the i-th sample, where C is the number of classes. The goal is to map each class label to an integer, so the transformation is defined as (3):

$$y_i^{\text{encoded}} = \text{LabelEncoder}\left(y_i\right), \tag{3}$$

where $y_i^{\text{encoded}} \in \{0, 1, \ldots, C-1\}$ represents the corresponding index of the label in a sorted list of unique classes.

The resulting encoded labels are stored in a vector $y^{\text{encoded}} \in \mathbb{R}^N$, where each y_i^{encoded} represents the integer value corresponding to the original class label.

One-Hot Encoding: While label encoding is a simple way of converting categorical labels into integers, for many classification tasks, especially in neural networks, it is more effective to represent each class label as a one-hot encoded vector. In one-hot encoding, each class label is represented as a vector where only the index corresponding to the label is 1, and all other indices are 0.

For a given encoded label $y_i^{\text{encoded}} \in \{0, 1, \ldots, C-1\}$, we convert it into a one-hot vector $y_i^{\text{onehot}} \in \mathbb{R}^C$. The one-hot encoding is given by (4):

$$y_i^{\text{onehot}} = [0, 0, \ldots, 1, \ldots, 0] \tag{4}$$

where the 1 appears at the index y_i^{encoded}, and all other positions are 0.

The entire matrix of one-hot encoded labels is represented as $Y^{\text{onehot}} \in \mathbb{R}^{N \times C}$, where each row y_i^{onehot} corresponds to the one-hot encoded vector for the i-th sample.

3.2.2 Hybrid WGAN-AE Model

The WGAN-AE is an upgraded model that combines the best features of autoencoders with GANs to offer superior attack detection in IoT networks. The model has two main modules: Autoencoder and Discriminator. In addition to that, the Wasserstein loss helps improve the training stability since the overall model learned to reconstruct the input data and distinguish between real and that which has been simulated. Mathematical formulation of the Hybrid WGAN-AE model is presented in this section.

A. Generator Architecture (Autoencoder)

A combination of an encoder and a decoder module constitutes the generator using the autoencoder structure. The autoencoder learns to accurately compress/input data into a low-dimensional representation, able to reconstruct the same input data from low-dimensionality space.

- Encoder: The encoder maps the input data x_i from the input space $\mathbb{R}^d$ to a lower-dimensional latent space. The encoder can be defined by a function $f_\theta : \mathbb{R}^d \to \mathbb{R}^z$, where θ represents the parameters of the encoder, and $z \in \mathbb{R}^z$ is the latent representation. Mathematically, the encoder can be expressed as:

  $$z_i = f_\theta(x_i),$$

  where z_i is the compressed representation of x_i.
- Decoder: The decoder is responsible for reconstructing the original input x_i from the latent vector z_i. The decoder function $g_\phi : \mathbb{R}^z \to \mathbb{R}^d$ is parameterized by ϕ, and the reconstruction $\hat{x}_i$ is given by:

  $$\hat{x}_i = g_\phi(z_i),$$

  where $\hat{x}_i$ is the reconstructed input.
  The loss function for the autoencoder is typically the reconstruction loss, which measures how well the decoder can approximate the original input x_i from the latent code z_i. The reconstruction loss is given by:

  $$L_{\text{recon}}(x_i, \hat{x}_i) = \|x_i - \hat{x}_i\|_2^2,$$

  where $\|\cdot\|_2^2$ denotes the squared Euclidean distance.

B. Discriminator Architecture

The discriminator is a neural network that distinguishes between real data (from the dataset) and fake data (generated by the autoencoder). It is defined as a binary classifier $D_\psi : \mathbb{R}^d \to [0,1]$, where $D_\psi(x_i) = 1$ indicates that x_i is real and $D_\psi(x_i) = 0$ indicates that x_i is fake.

For the input data x_i, the discriminator outputs a scalar $D_\psi(x_i)$ representing the probability that x_i is real. The discriminator is trained to maximize this probability when the data is real and minimize it when the data is fake. Thus, the loss for the discriminator can be described as (5):

$$L_{\text{disc}}(x_i, \hat{x}_i) = -\mathbb{E}_{x \sim P_{\text{real}}}\left[\log D_\psi(x)\right] - \mathbb{E}_{\hat{x} \sim P_{\text{gen}}}\left[\log\left(1 - D_\psi(\hat{x})\right)\right], \tag{5}$$

where P_{real} indicates the distribution of real data and P_{gen} represents the distribution of generated data.

The goal of the discriminator is to correctly classify real data as 1 and fake data as 0, minimizing the above loss function.

C. Wasserstein Loss for WGAN-AE

A key component of the Hybrid WGAN-AE is the Wasserstein loss, which is used to stabilize the training process, and to generate better samples.

The Wasserstein loss [24] for the discriminator is given by (6):

$$L_{\text{Wasserstein}}\left(x_i, \hat{x}_i\right) = \mathbb{E}_{x\sim P_{\text{real}}}\left[D_\psi(x)\right] - \mathbb{E}_{\hat{x}\sim P_{\text{gen}}}\left[D_\psi(\hat{x})\right]. \tag{6}$$

This loss function trains the discriminator to give high values to real data and low values to the generated data, thus trying to increase the distance between the two distributions. The Wasserstein loss has smoother gradients and solves the vanishing gradient problem that is typical for standard GANs.

The generator is trained to minimize the Wasserstein loss in the opposite direction, i.e., to minimize $-L_{\text{Wasserstein}}$, which ensures that the generated data distribution approaches the real data distribution.

D. Combined Model

In the Hybrid WGAN-AE, the combined model is trained alongside both the generator and the discriminator. The purpose of the combined model is to train the generator to create data that looks realistic enough to fool the discriminator. Nevertheless, the generator tries to minimize the reconstruction error as well so that the produced data is both realistic and consistent with the input data.

The total loss for the combined model is a weighted sum of the reconstruction loss and the Wasserstein loss. The combined loss can be written as follows:

$$L_{\text{combined}} = \lambda_{\text{recon}} L_{\text{recon}}\left(x_i, \hat{x}_i\right) + \lambda_{\text{Wasserstein}} L_{\text{Wasserstein}}\left(x_i, \hat{x}_i\right),$$

where λ_{recon} and $\lambda_{\text{Wasserstein}}$ are hyperparameters controlling the relative importance of the reconstruction loss and the Wasserstein loss. The generator is trained to minimize this combined loss. The workflow of WGAN-AE is summarized in Algorithm 1.

Algorithm 1: Workflow of the proposed hybrid WGAN-AE model

Require: Dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$
Ensure: Trained Generator (Autoencoder) and Discriminator
1: **Data Preprocessing:**
2: Standardize features:
3: $\mu_j = \frac{1}{N}\sum_{i=1}^{N} x_{ij}$
4: $\sigma_j = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_{ij} - \mu_j)^2}$
5: $x_j^{\text{scaled}} = \frac{x_j - \mu_j}{\sigma_j}$
6: Encode labels using one-hot encoding.
7: Initialize model parameters: θ (encoder), φ (decoder), ψ (discriminator)
8: **Generator (Autoencoder) Architecture:**
9: Encode input: $z_i = f_\theta(x_i)$

(Continued)

Algorithm 1 (continued)

Decode output: $\hat{x}_i = g_\phi(z_i)$
Compute reconstruction loss:
$L_{\text{recon}} = \|x_i - \hat{x}_i\|_2^2$
Discriminator Architecture:
Compute real data score: $D_\psi(x_i)$
Compute fake data score: $D_\psi(\hat{x}_i)$
Compute discriminator loss:
$L_{\text{disc}} = -\mathbb{E}_{x\sim P_{\text{real}}}[\log D_\psi(x)] - \mathbb{E}_{\hat{x}\sim P_{\text{gen}}}[\log(1 - D_\psi(\hat{x}))]$
Compute Wasserstein loss:
$L_{\text{Wasserstein}} = \mathbb{E}_{x\sim P_{\text{real}}}[D_\psi(x)] - \mathbb{E}_{\hat{x}\sim P_{\text{gen}}}[D_\psi(\hat{x})]$
Compute combined loss:
$L_{\text{combined}} = \lambda_{\text{recon}} L_{\text{recon}} + \lambda_{\text{Wasserstein}} L_{\text{Wasserstein}}$
return Trained Generator (Autoencoder) and Discriminator

E. Training Procedure

The training procedure alternates between updating the discriminator and the generator:

1. Discriminator Update: The discriminator is trained to distinguish between real and fake data by minimizing L_{disc} . The real data samples x_i are drawn from the dataset, and the fake data samples $\hat{x}_i$ are generated by the autoencoder.
2. Generator Update: The generator is updated to minimize the combined loss $L_{\text{combined,}}$ which includes both the reconstruction error and the Wasserstein loss. This is done by training the generator to fool the discriminator and simultaneously reconstruct the input data accurately.

The discriminator's parameters ψ are updated using a standard optimization algorithm, while the generator's parameters θ and ϕ are updated jointly. The training process is summarized in the Algorithm 2.

Algorithm 2: Training process of the proposed hybrid WGAN-AE

Require: Dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$, learning rates η_1, η_2, batch size B, epochs T
Ensure: Trained Generator (Autoencoder) and Discriminator
Initialize parameters: θ (encoder), φ (decoder), ψ (discriminator)
for $t = 1$ to T **do**
Sample mini-batch $\{x_i\}$ from dataset.
Update Discriminator:
Compute $D_\psi(x_i)$ and $D_\psi(\hat{x}_i)$
Compute Wasserstein loss:
$L_{\text{Wasserstein}} = \mathbb{E}_{x\sim P_{\text{real}}}[D_\psi(x)] - \mathbb{E}_{\hat{x}\sim P_{\text{gen}}}[D_\psi(\hat{x})]$
Update ψ:
$\psi \leftarrow \psi - \eta_1 \nabla_\psi L_{\text{Wasserstein}}$
Update Generator (Autoencoder):
Encode input: $z_i = f_\theta(x_i)$
Decode output: $\hat{x}_i = g_\phi(z_i)$

(Continued)

Algorithm 2 (continued)

Compute combined loss:
$L_{\text{combined}} = \lambda_{\text{recon}} ||x_i - \hat{x}_i||_2^2 + \lambda_{\text{Wasserstein}} L_{\text{Wasserstein}}$
Update θ, ϕ:
$(\theta, \phi) \leftarrow (\theta, \phi) - \eta_2 \nabla_{\theta,\phi} L_{\text{combined}}$
end for
return Trained Generator (Autoencoder) and Discriminator

4 Experiments and Results

The proposed WGAN-AE architecture is implemented and evaluated in the Google Colab Pro platform. To ensure the optimal training of WGAN-AE, we selected the range of suitable hyperparameters through the hit and trial method. The customized and default hyperparameters utilized in training are presented in Tables 2 and 3, respectively. The following provides a brief discussion of the experimental procedures and an in-depth analysis of experimental outcomes.

Table 2: Customized hyperparameters for the training of proposed WGAN-AE

Parameter	Value
Encoding dimension	14
Discriminator layers	128, 64 neurons (ReLU activation)
Generator activation	ReLU (encoding), Sigmoid (decoding)
Loss function (GAN)	Wasserstein Loss
Loss function (Final Model)	Mean squared error, Categorical Crossentropy (loss weights: 0.5, 0.5)
Optimizer	Adam
Batch size	256
Epochs (GAN Training)	10
Epochs (K-Fold Training)	5
K-Fold splits	5
Random seed	42

Table 3: Default hyperparameters for the training of proposed WGAN-AE

Parameter	Default value
Learning rate (Adam Optimizer)	0.001
Beta1 (Adam Optimizer)	0.9
Beta2 (Adam Optimizer)	0.999
Epsilon (Adam Optimizer)	1e-7
Weight initialization (Dense Layers)	Xavier initialization

4.1 Performance Evaluation with 5G-NIDD Dataset

The 5G-NIDD dataset is an important benchmark for evaluating the performance of intrusion detection systems in next-generation 5G-enabled IoT networks, characterized by speed, low latency, and a variety of threats. The following presents a brief discussion of the experimental results.

4.1.1 Cross-Validation Performance

To ensure robustness, the model was subjected to five-fold cross validation to prevent overfitting on a single dataset. The five-fold cross-validation results are shown in Fig. 4. The experimental outcomes delivered quite impressive Precision-Recall AUC scores ranging from 99.79% to 99.91%, with an average of 99.87%. This is because the model is very precise, with low false positive and false negative rates for high accuracy. The scores for the Balanced Accuracy ranged from 98.79% to 98.97% with an average of 98.91%. These cross validation results show that the model is highly generalizable with stable performance across different data splits.

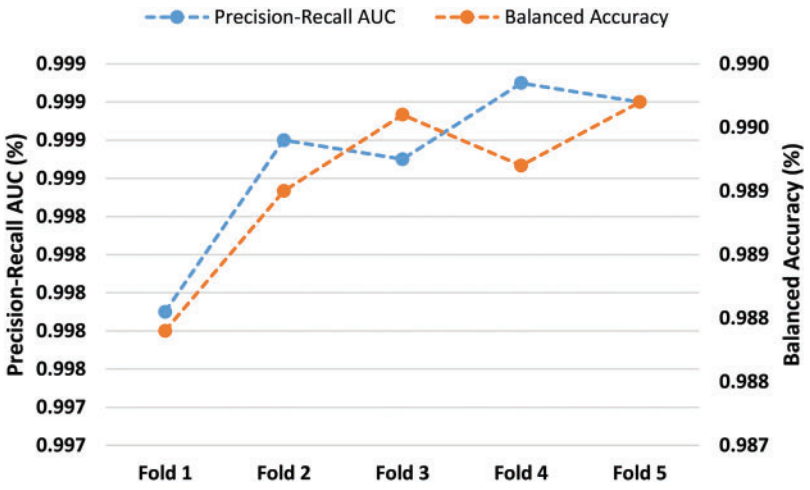

Figure 4: 5-fold cross validation with 5G-NIDD dataset

4.1.2 Test Set Performance

To further assess the model's effectiveness, it was evaluated on an independent test set, where it achieved a notable performance score. The accuracy was 97.35%, with precision, recall and F1-score all being very close to each other at 97.39%, 97.35%, and 97.35, respectively. These almost equal values across the different metrics indicate that the model has a good center that helps it avoid false positives and negatives. The high F1 score further confirms that both precision and recall are good, so the model does not give many false positives while also detecting malicious traffic effectively.

4.1.3 Multiclass Performance Analysis

The confusion matrix presented in Fig. 5 provides deeper insights into the model's classification performance. It shows that the model made few misclassifications, particularly for well-defined attack patterns such as UDP Flood, HTTP Flood, and SYN Scan. However, some minor misclassifications were observed between attacks with similar characteristics, such as SYN Flood and TCP Connect Scan, where the model occasionally confused connection-based attacks due to their similar network behavior profiles. Despite these minor misclassifications, the overall error rate was low, reinforcing the model's high reliability. Fig. 6 presents a detailed multiclass performance evaluation.

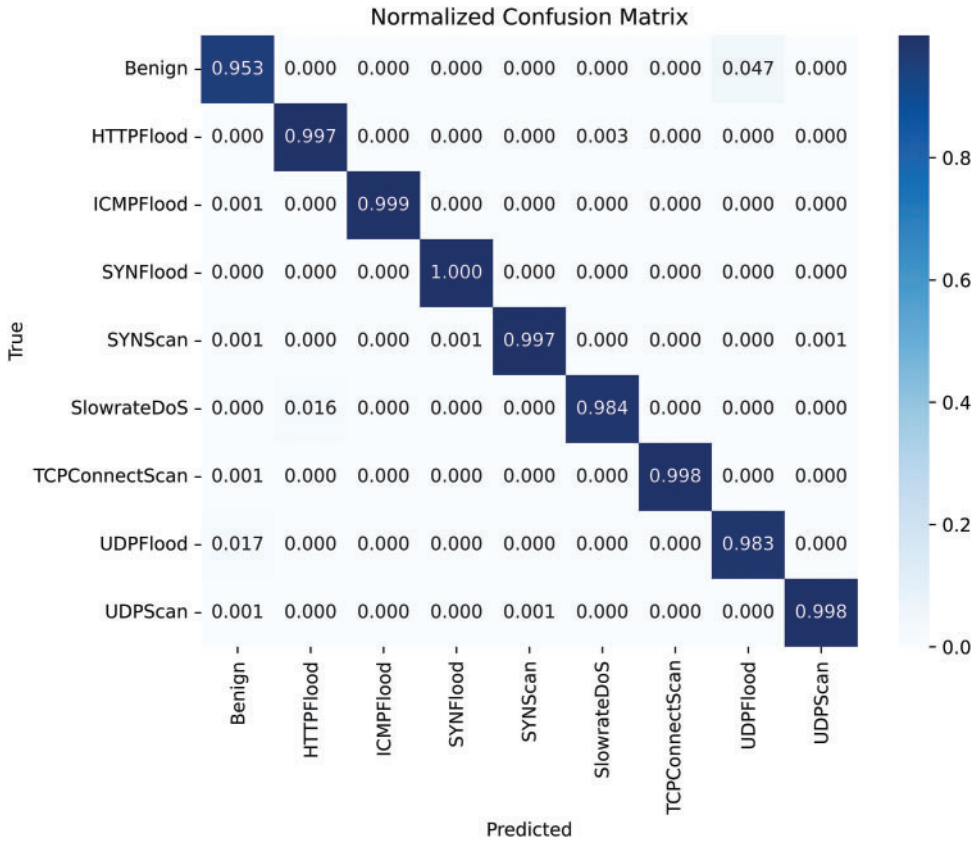

Figure 5: Confusion matrix for 5G-NIDD dataset

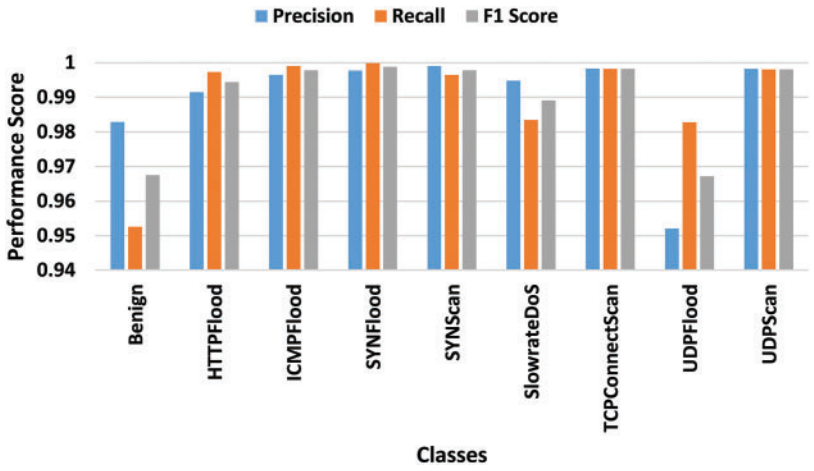

Figure 6: Multiclass performance evaluation with 5G-NIDD dataset

4.1.4 Computational Performance Analysis

The computational performance analysis of the proposed WGAN-AE model on the 5G-NIDD dataset demonstrates efficient processing capabilities. The training phase was completed in 59.20913 s, indicating that the model can handle complex data with moderate computational requirements. During the inference stage, the model processed the entire test set in 81.51074 s, reflecting its capacity to analyze and generate predictions

efficiently. The per-sample latency was measured at 0.06704 ms, ensuring minimal delay during inference. Moreover, the throughput of the model was recorded at 14, 916.93066 samples per second, highlighting its capability to process a large volume of data in heterogeneous IoT networks.

4.1.5 Attack Detection Time Analysis

In real-world applications, fast detection of attacks is crucial to mitigate threats in real time. The proposed model demonstrated impressive low-latency performance across various attack types, processing most attacks within 0.06 to 0.08 milliseconds. This ensures that it can quickly identify and respond to malicious activities without causing significant delays. However, the ICMP Flood attack took slightly longer to detect at 0.29605 ms, likely due to its bursty nature and the larger packet sizes involved, which required additional computational resources. Despite this, all other attack detection times remained within sub-millisecond latencies, confirming that the model is well-suited for real-time intrusion detection in 5G-enabled IoT networks. Fig. 7 illustrates each class's attack detection time.

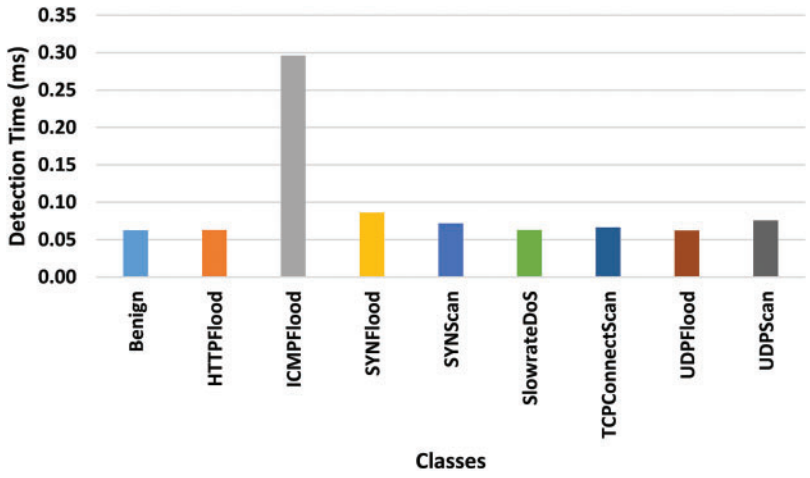

Figure 7: Detection time of each individual class in 5G-NIDD dataset

4.1.6 Model Size and Deployment Feasibility

Considering the limited storage and computational resources available on many IoT devices, the model's memory footprint is a key factor for deployment feasibility. The proposed model has a compact size of just 60.24 kB, making it highly efficient for deployment on resource-constrained IoT devices. The smaller memory footprint, high detection accuracy, and low latency make it an ideal choice for deployment in real-world IoT networks.

4.2 Performance Evaluation with IDSIoT2024 Dataset

The IDSIoT2024 dataset contains diverse IoT traffic data to evaluate IDSs in dynamic and resource-constrained IoT networks. The following provides a detailed analysis of experimental results with the IDS IoT 2024 dataset.

4.2.1 Cross-Validation Performance

The five-fold cross-validation results are presented in Fig. 8. The Precision-Recall AUC scores for the five folds were very high, with an average of 94.09%, ranging from 91.81% to 95.66%. This shows that the

model is capable of a good precision-recall tradeoff, i.e., it can avoid many false positives and false negatives. The Balanced Accuracy values were also very good, with a range of 88.54%–91.59% and an average of 90.53%. These results show that the model performs reliably in distinguishing attack and benign traffic without bias and thus is likely to be deployable in a generalizable manner across IoT networks.

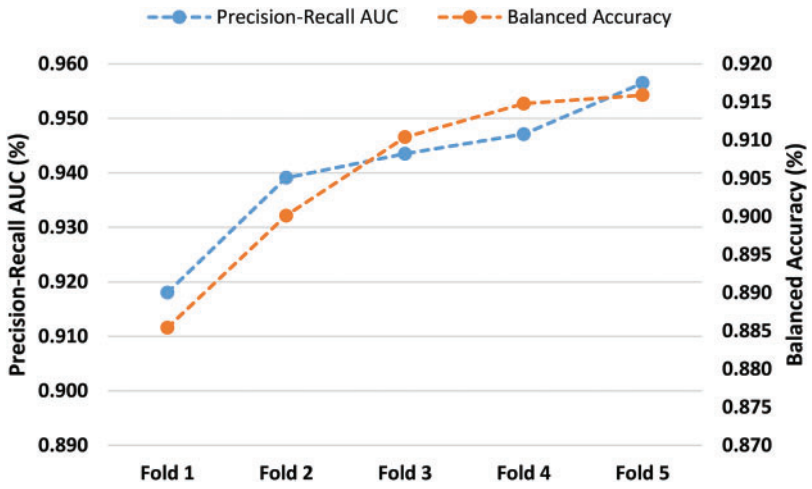

Figure 8: 5-fold cross validation with IDSIoT2024 dataset

4.2.2 Test Set Performance

After training the model using the training dataset and validating its efficiency using the validation set, it performed exceptionally on an independent test set with high accuracy of 98.38% to distinguish between normal and malicious traffic. All other metrics like precision, recall and F1 scores were also very close to each other, with values of 98.34%, 98.38% and 98.27%, respectively. This indicates that the model performs well without being too sensitive or ignoring real anomalies. Furthermore, the decision-making process of the model is straightforward, making it easy to interpret and trust the results.

4.2.3 Multiclass Performance Analysis

The confusion matrix in Fig. 9 gives a more accurate view of the model's classification accuracy. When the multi-class evaluation was performed, the model's performance was found to be consistent across all the other classes except the 'Injection' class. This means that although the model is very good at identifying different attacks, there could be some difficulties in distinguishing between some of the attacks, especially Injection attacks, which may have attack patterns that are similar to those of other malicious activities. The low performance in this case can be ascribed to the characteristics of Injection attacks which are often quiet and their traffic is not easily distinguishable. Nonetheless, the general multiclass performance is good which indicates that the model is well positioned to deal with different kinds of attacks. The multiclass performance is detailed in Fig. 10.

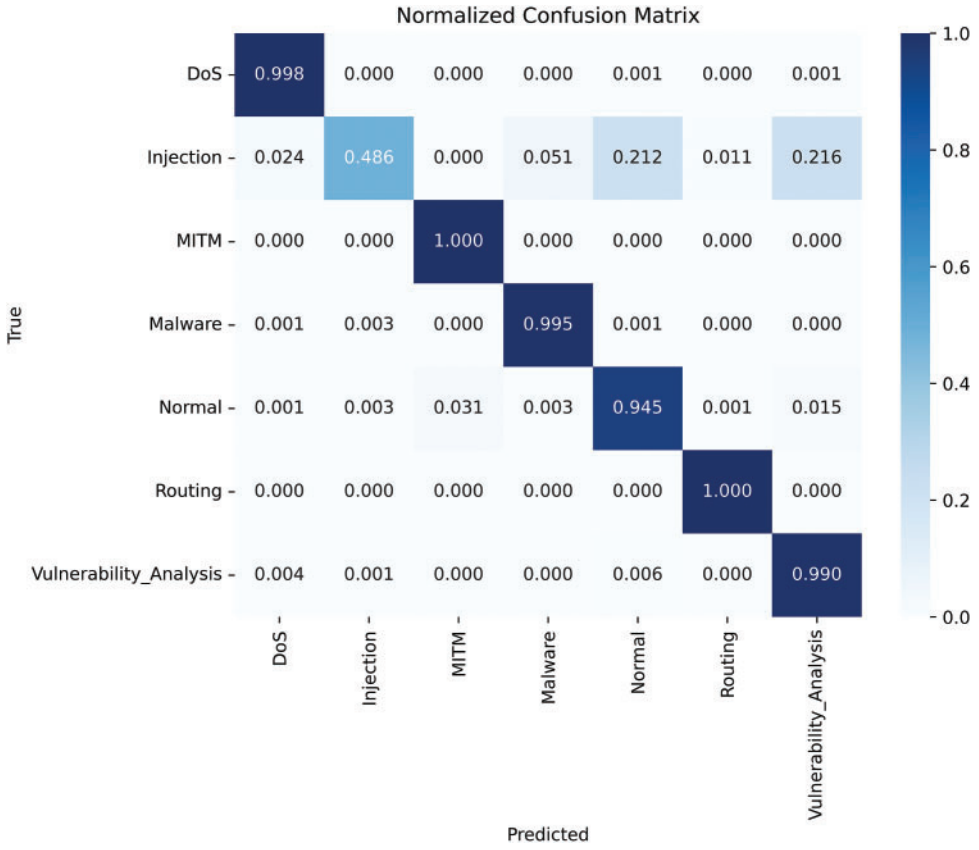

Figure 9: Confusion matrix for IDSIoT2024 dataset

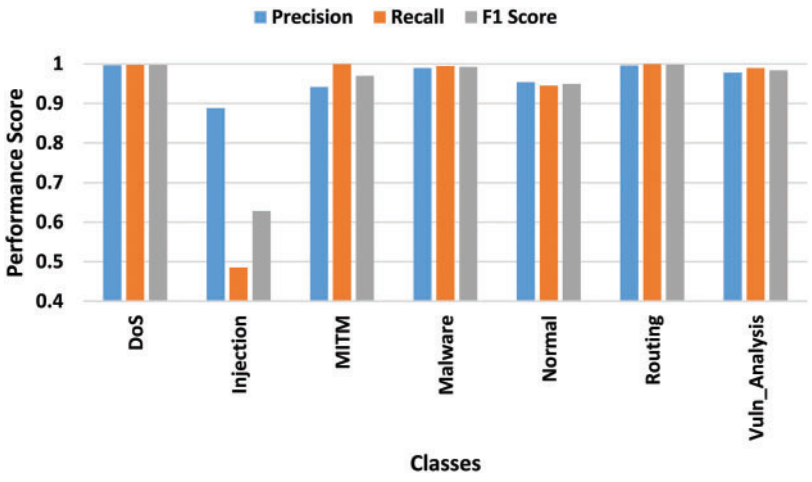

Figure 10: Multiclass performance evaluation with IDSIoT2024 dataset

4.2.4 Computational Performance Analysis

For the IDSIoT2024 dataset, the proposed WGAN-AE model exhibited a significantly lower training time of 10.67866 s, suggesting that the model adapts efficiently to this dataset. The inference time for

the complete test set was remarkably fast at 6.43554 s, emphasizing the model's ability to handle large-scale IoT data effectively. The latency per sample was recorded at 0.06684 ms, demonstrating minimal delay during prediction. Additionally, the throughput achieved was 14,960.67771 samples per second, reflecting a high data processing rate, which is crucial for real-time IoT applications. These results affirm that the WGAN-AE model is computationally efficient across diverse datasets, making it well-suited for high-throughput environments.

4.2.5 Attack Detection Time Analysis

The model demonstrated impressive efficiency with attack detection times, processing most attacks in sub-millisecond times, ranging from 0.05841 to 0.11560 ms. The shortest detection times were observed for normal traffic and attacks like DoS and Routing, indicating the model's ability to detect these traffic types quickly. The slightly longer detection times for more complex attacks, such as Injection (0.10729 ms) and MITM (0.11560 ms), still remained well within acceptable thresholds for real-time monitoring. These results highlight the model's suitability for use in environments where prompt threat detection is essential. Fig. 11 illustrates each class's attack detection time.

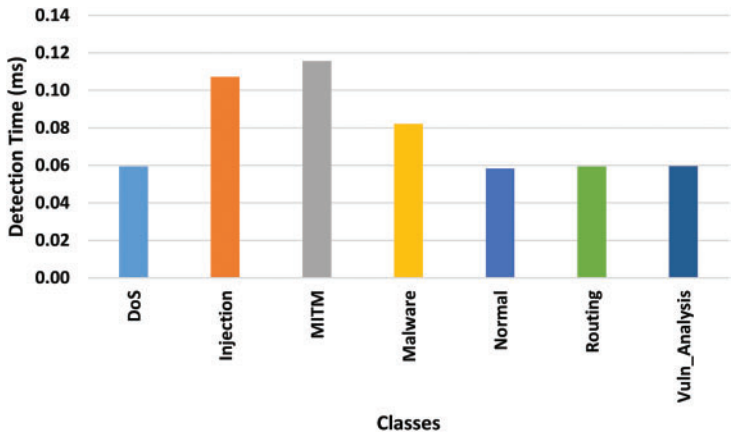

Figure 11: Detection time of each individual class in IDSIoT2024 dataset

4.2.6 Model Size and Deployment Feasibility

The model's compact size of 61.84 kB makes it highly suitable for deployment in resource-constrained IoT environments. This small memory footprint ensures that the model can be easily integrated into IoT devices with limited computational resources. The model's lightweight nature, high detection accuracy, and low detection latency demonstrate its potential for deployment in real-world IoT networks, where both efficiency and security are critical. Given the increasing demand for effective and resource-efficient intrusion detection in IoT systems, this model offers a promising solution for ensuring the security of IoT devices.

4.3 Performance Comparison of Proposed WGAN-AE with State-of-the-Art GAN Variants

To analyze the efficacy of the proposed WGAN-AE, we compared the performance with state-of-the-art GAN variants, including vanilla GAN, conditional GAN, least square GAN, Information Maximizing Generative Adversarial Networks (info GAN), and boundary equilibrium GAN. To ensure a fair comparison, we implemented all these GAN models on similar experimentation platforms with similar datasets. The following presents a detailed comparative analysis.

4.3.1 Detection Rate and False Alarm Rate

The WGAN-AE consistently outperforms other GAN variants in detection accuracy while maintaining lower false alarm rates across different attack classes. Table 4 presents comparative evaluation results for the 5G-NIDD dataset. The proposed scheme achieved a remarkable 100% detection rate for ICMPFlood attacks with zero false alarms, a stark contrast to Vanilla GAN's 29.69% detection rate. Similarly, for SYNFlood and SYNScan attacks, WGAN-AE achieves near-perfect detection, significantly surpassing alternatives such as Least Squares GAN and Boundary Equilibrium GAN. The improvement is also evident in benign traffic classification, where WGAN-AE attains a detection rate of 95.84% with a reduced false alarm rate of 1.93%, demonstrating its robustness in distinguishing normal and attack traffic.

Table 4: Comparative analysis of detection rate (DR) vs. false alarm rate (FAR) for 5GNIDD Dataset (all values are in percentage, %)

Class	Vanilla GAN		Conditional GAN		Least Squares GAN		Info GAN		Boundary Equilibrium GAN		WGAN-AE (Proposed Scheme)	
	DR	FAR	DR	FAR	DR	FAR	DR	FAR	DR	FAR	DR	FAR
Benign	87.094	6.529	82.563	8.047	79.527	6.197	93.227	3.683	87.942	4.784	95.847	1.927
HTTPFlood	97.532	0.635	98.203	1.074	97.016	0.753	98.455	0.338	97.566	0.517	99.471	0.135
ICMPFlood	29.697	0.002	0.000	0.000	0.087	0.000	98.788	0.002	61.385	0.005	100.00	0.000
SYNFlood	85.876	0.019	82.636	0.028	83.870	0.025	86.092	0.010	85.639	0.011	99.928	0.003
SYNScan	95.375	0.019	92.616	0.050	92.840	0.043	99.606	0.008	97.296	0.016	99.776	0.006
SlowrateDoS	90.687	0.331	84.521	0.239	88.885	0.375	94.907	0.181	92.338	0.300	98.009	0.065
TCPConnectScan	98.499	0.188	96.225	0.252	96.848	0.245	98.998	0.116	98.768	0.160	99.771	0.002
UDPFlood	89.609	8.205	87.211	11.147	90.233	13.147	94.137	4.249	92.384	7.627	96.912	2.604
UDPScan	93.015	0.019	87.583	0.044	85.924	0.019	99.453	0.005	96.259	0.020	99.642	0.002

Table 5 presents the comparative analysis for the IDSIoT2024 dataset. A similar trend is observed in the IDSIoT2024 dataset, where WGAN-AE exhibited high detection rates for Routing (99.35%) and DoS (99.05%) attacks, outperforming all other models. The false alarm rate remains consistently low, reinforcing its reliability for real-world deployment. However, a noticeable weakness emerges in detecting Injection attacks, where it underperformed compared to Information Maximising Generative Adversarial Networks (InfoGAN).

Table 5: Comparative analysis of detection rate (DR) vs. false alarm rate (FAR) for IDSIoT2024 Dataset (all values are in percentage, %)

Class	Vanilla GAN		Conditional GAN		Least squares GAN		Info GAN		Boundary equilibrium GAN		WGAN-AE (Proposed Scheme)	
	DR	FAR	DR	FAR	DR	FAR	DR	FAR	DR	FAR	DR	FAR
DoS	94.093	8.451	97.008	4.893	96.200	14.357	97.993	1.923	95.230	6.198	99.053	0.565
Injection	34.063	0.078	43.125	0.073	35.859	0.061	52.656	0.118	36.484	0.058	40.156	0.093
MITM	100.000	0.340	100.00	0.340	100.00	0.340	100.000	0.340	100.00	0.340	100.00	0.340
Malware	99.180	0.404	99.410	0.210	99.380	0.358	99.190	0.072	99.380	0.298	99.420	0.294
Normal	82.980	0.935	88.520	0.866	83.510	1.159	91.980	0.935	84.880	0.794	91.030	0.688
Routing	65.620	2.174	78.330	0.811	33.280	0.806	92.090	0.422	77.100	1.781	99.350	0.010
Vuln_Analysis	95.510	1.514	97.610	1.332	94.825	1.713	97.590	1.085	95.920	1.401	98.365	1.205

4.3.2 Computational Efficiency and Resource Utilization

One of the promising features of WGAN-AE is its efficient training process, which is significantly faster than traditional GANs. The computational efficiency and resource utilization comparison are presented in Tables 6 and 7 for 5G-NIDD and IDSIoT2024 datasets, respectively. On the 5G-NIDD dataset, the training time is reduced to 59.2 s, compared to InfoGAN's 214.9 s and Vanilla GAN's 208.2 s. This efficiency stems from incorporating an autoencoder, which compresses input data before training, substantially reducing computational complexity. A similar advantage is observed in the IDSIoT2024 dataset, where the training time is just 10.67 s, making WGAN-AE faster training models among its peers. Another notable advantage of WGAN-AE is its remarkably small model size. The model requires only 0.060 MB for 5G-NIDD and 0.061 MB for IDSIoT2024, making it an excellent choice for deployment in resource-constrained IoT environments.

Table 6: Comparative analysis of training cost, inferencing time, latency, throughput, and model size for 5GNIDD dataset

GAN variants	Training cost (sec)	Inferencing time (sec)	Latency (ms)	Throughput (Samples/sec)	Model size (MBs)
Vanilla GAN	208.23392	61.12484	0.05027	19891	0.78463
Conditional GAN	112.46435	60.25724	0.04956	20178	0.78463
Least squares GAN	155.00871	60.92937	0.05011	19955	8.65162
Info GAN	214.93845	61.21826	0.05035	19861	0.78463
Boundary equilibrium GAN	208.44314	71.86630	0.05911	16918	0.78463
WGAN-AE	59.20913	81.51074	0.06704	14916	0.06024

Table 7: Comparative analysis of training cost, inferencing time, latency, throughput, and model size for IDSIoT2024 dataset

GAN variants	Training cost (sec)	Inferencing time (sec)	Latency (ms)	Throughput (Samples/sec)	Model size (MBs)
Vanilla GAN	27.15654	5.66325	0.05882	17000	0.79781
Conditional GAN	19.73150	5.18062	0.05381	18584	0.79781
Least squares GAN	32.70655	5.26421	0.05468	18289	8.70435
Info GAN	26.47308	4.95029	0.05142	19449	0.79781
Boundary equilibrium GAN	24.77814	5.02102	0.05215	19175	0.79781
WGAN-AE	10.67866	6.43554	0.06684	14960	0.06184

4.4 Trade-offs and Limitations: A Balanced Perspective on WGAN-AE's Promising Performance

The WGAN-AE model demonstrates remarkable performance in terms of detection accuracy and false alarm reduction, significantly outperforming state-of-the-art GAN variants across diverse attack classes. While a detailed evaluation reveals some trade-offs in inferencing time and throughput, these differences are relatively minor compared to the notable improvements WGAN-AE brings in detection performance.

4.4.1 Latency and Inferencing Time: Minimal Trade-Offs for Superior Detection

A closer analysis of the inferencing time and latency highlights that although WGAN-AE incurs a slightly higher latency (0.067 ms for 5G-NIDD and 0.066 ms for IDSIoT2024) compared to InfoGAN and Vanilla GAN (around 0.051 ms), the difference is negligibly small in practical scenarios. This minimal latency overhead is a small price for achieving significantly higher detection rates and lower false alarm rates. In real-world intrusion detection environments, where accurate and reliable attack classification is paramount, this marginal increase in latency does not compromise the system's overall responsiveness.

4.4.2 Throughput: Prioritizing Accuracy over Speed in High-Stakes Scenarios

Similarly, while WGAN-AE demonstrates slightly lower throughput (around 14, 916 samples/sec for 5G-NIDD and 14, 960 samples/sec for IDSIoT 2024) compared to InfoGAN and Conditional GAN (which process over 19, 000 samples/sec), this trade-off is more than compensated for by the model's superior detection performance. Although important in high-speed environments, it becomes secondary in scenarios where accuracy and reliability are critical. For instance, in mission-critical IoT or 5G networks, ensuring that malicious traffic is identified with near-zero false alarms is more desirable than marginally higher processing speed.

4.4.3 Detection Superiority and False Alarm Reduction: WGAN-AE's Competitive Edge

The strength of WGAN-AE lies in its ability to consistently achieve higher detection rates across a wide range of attack types, including difficult-to-detect threats such as ICMPFlood, SYNFlood, TCPConnectScan, and UDPScan, while simultaneously maintaining a significantly lower false alarm rate. For example, on the 5G-NIDD dataset, WGAN-AE achieves a 100% detection rate for ICMPFlood attacks with a 0% false alarm rate, outperforming all other models. Even for complex attack types in the IDSIoT2024 dataset, WGAN-AE maintains exceptional performance, highlighting its robustness and reliability in detecting both known and emerging threats.

4.5 Future Directions: Enhancing WGAN-AE for Greater Efficiency

While WGAN-AE has already set a high standard in intrusion detection, a few strategic enhancements could further refine its performance regarding inferencing time and throughput. The following two recommendations can help address these minor trade-offs:

4.5.1 Model Pruning and Lightweight Architectures for Faster Inferencing

Model pruning and lightweight architectures can be employed to minimize inferencing time and latency without sacrificing detection accuracy. Pruning reduces model complexity by removing redundant connections and neurons, leading to a lighter and faster network while retaining essential feature representations. Additionally, incorporating quantization techniques can further reduce model size and computation requirements, making WGAN-AE more efficient for real-time applications. This approach can maintain the model's detection superiority while ensuring faster inferencing in large-scale or latency-sensitive environments.

4.5.2 Parallel Processing and Distributed Inference for Higher Throughput

Implementing parallel processing techniques and distributed inference frameworks can significantly improve performance in high-speed networks, enhancing throughput and enabling real-time intrusion detection. WGAN-AE can process a larger volume of samples concurrently by partitioning incoming network traffic across multiple computational nodes, thereby increasing overall throughput. Additionally,

leveraging edge-cloud hybrid architectures can offload preliminary anomaly detection to edge devices, reducing the computational burden on central servers while maintaining accuracy and reliability.

5 Conclusion

This paper proposed a hybrid framework for an efficient IDS for IoT networks using GAN and autoencoder architectures. The proposed WGAN-AE successfully identified a range of cyberattacks with higher accuracy. The performance of the designed IDS framework was evaluated using two open source datasets, 5GNIDD and IDSIoT2024. The experimental outcomes confirm the higher attack detection accuracy in both 5-fold cross-validation scenarios and with respect to independent testing data. The microseconds attack detection time for each class and the low memory footprint makes it suitable for deployment in resource constrained IoT devices and networks.

Acknowledgement: The authors extend their appreciation to the Deanship of Research and Graduate Studies at King Khalid University for funding this work through Large Group Project under grant number (RGP.2/245/46). This work is funded by Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2025R760), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia. The research team thanks the Deanship of Graduate Studies and Scientific Research at Najran University for supporting the research project through the Nama'a program, with the project code NU/GP/SERC/13/352-1.

Funding Statement: The authors extend their appreciation to the Deanship of Research and Graduate Studies at King Khalid University for funding this work through Large Group Project under grant number (RGP.2/245/46). This work is funded by Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2025R760), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia. The research team thanks the Deanship of Graduate Studies and Scientific Research at Najran University for supporting the research project through the Nama'a program, with the project code NU/GP/SERC/13/352-1.

Author Contributions: Mohammed S. Alshehri: Writing an original draft, Visualization, Validation, Software, Project administration, Methodology, Investigation, Formal analysis, Conceptualization. Oumaima Saidani: Writing an original draft, Visualization, Methodology, Investigation, Formal analysis, and Conceptualization. Wajdan Al Malwi: Writing, review & editing, Writing an original draft, Visualization, Validation. Fatima Asiri: Writing, review & editing, Visualization, Validation. Shahid Latif: Writing an original draft, Software, Methodology, Formal analysis. Aizaz Ahmad Khattak: Review & editing, Visualization, Methodology. Jawad Ahmad: Review & editing, Software, Project administration. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: We are happy to share the processed datasets and Jupyter Notebooks of the proposed scheme for research purposes upon request, subject to the approval of our research group.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Mehedi ST, Anwar A, Rahman Z, Ahmed K, Islam R. Dependable intrusion detection system for IoT: a deep transfer learning based approach. IEEE Trans Indus Inform. 2022;19(1):1006–17. doi:10.1109/TII.2022.3164770.
2. Shafiq M, Gu Z, Cheikhrouhou O, Alhakami W, Hamam H. The rise of "Internet of Things": review and open research issues related to detection and prevention of IoT-based security attacks. Wirel Commun Mob Comput. 2022;2022(1):8669348. doi:10.1155/2022/8669348.
3. Jarwar MA, FREng JWC, Ali S. Modelling industrial IoT security using ontologies: a systematic review. IEEE Open J Commun Soc. 2025;6(3):2792–821. doi:10.1109/OJCOMS.2025.3532224.

4. Latif S, Huma Z, Jamal SS, Ahmed F, Ahmad J, Zahid A, et al. Intrusion detection framework for the internet of things using a dense random neural network. IEEE Trans Indus Inform. 2021;18(9):6435–44. doi:10.1109/TII.2021.3130248.
5. Alwaisi Z, Soderi S. Towards robust IoT defense: comparative statistics of attack detection in resource-constrained scenarios. In: EAI International Conference on Body Area Networks; 2024 Feb 4–5; Milan, Italy. p. 272–91.
6. 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 of Things. 2024;26(7):101212. doi:10.1016/j.iot.2024.101212.
7. Lin R, Qiu H, Wang J, Zhang Z, Wu L, Shu F. Physical layer security enhancement in energy harvesting-based cognitive internet of things: a GAN-powered deep reinforcement learning approach. IEEE Internet of Things J. 2024;11(3):4899–913. doi:10.1109/JIOT.2023.3300770.
8. Karthika S, Durgadevi M. Generative Adversarial Network (GAN): a general review on different variants of GAN and applications. In: 2021 6th International Conference on Communication and Electronics Systems (ICCES). Coimbatre, India: IEEE; 2021. p. 1–8.
9. Fetaya E, Jacobsen JH, Grathwohl W, Zemel R. Understanding the limitations of conditional generative models. arXiv:190601171. 2019.
10. Cabezon Pedroso T, Ser JD, Díaz-Rodríguez N. Capabilities, limitations and challenges of style transfer with CycleGANs: a study on automatic ring design generation. In: International Cross-Domain Conference for Machine Learning and Knowledge Extraction; 2022 Aug 23–26; Vienna, Austria: Springer. p. 168–87.
11. Hasan MN, Jan SU, Koo I. Wasserstein GAN-based digital twin-inspired model for early drift fault detection in wireless sensor networks. IEEE Sens J. 2023;23(12):13327–39. doi:10.1109/JSEN.2023.3272908.
12. Cai Z, Du H, Wang H, Zhang J, Si Y, Li P. One-dimensional convolutional wasserstein generative adversarial network based intrusion detection method for industrial control systems. Electronics. 2023;12(22):4653. doi:10.3390/electronics12224653.
13. Alrayes FS, Zakariah M, Amin SU, Khan ZI, Helal M. Intrusion detection in IoT systems using denoising autoencoder. IEEE Access. 2024;12:122401–25.
14. Boppana TK, Bagade P. GAN-AE: an unsupervised intrusion detection system for MQTT networks. Eng Appl Artif Intell. 2023;119(11):105805. doi:10.1016/j.engappai.2022.105805.
15. 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(15):122198. doi:10.1016/j.eswa.2023.122198.
16. de Araujo-Filho PF, Kaddoum G, Campelo DR, Santos AG, Macêdo D, Zanchettin C. Intrusion detection for cyber-physical systems using generative adversarial networks in fog environment. IEEE Internet of Things J. 2020;8(8):6247–56. doi:10.1109/JIOT.2020.3024800.
17. Zeghida H, Boulaiche M, Chikh R, Bamhdi AM, Barros ALB, Zeghida D, et al. Enhancing IoT cyber attacks intrusion detection through GAN-based data augmentation and hybrid deep learning models for MQTT network protocol cyber attacks. Cluster Comput. 2025;28(1):58. doi:10.1007/s10586-024-04752-5.
18. Das S, Majumder A, Namasudra S, Singh A. Intrusion detection using CTGAN and lightweight neural network for Internet of Things. Expert Syst. 2025;42(2):e13793. doi:10.1111/exsy.13793.
19. Wang H, Kandah F, Mendis T, Medury L. Clustering-based intrusion detection system meets multi-critics generative adversarial networks. IEEE Internet Things J. 2025. doi:10.1109/JIOT.2025.3533918.
20. Dong B, Wang H, Luo R. MasqueradeGAN-GP: a generative adversarial network framework for evading black-box intrusion detection systems. Internet Technol Lett. 2025;16(8):e640. doi:10.1002/itl2.640.
21. Brabin DD, Kumar KK, Sunitha T. Strengthening security in IoT-based smart cities utilizing cycle-consistent generative adversarial networks for attack detection and secure data transmission. Peer Peer Netw Appl. 2025;18(2):79. doi:10.1007/s12083-024-01838-0.
22. Samarakoon S, Siriwardhana Y, Porambage P, Liyanage M, Chang SY, Kim J, et al. 5G-NIDD: a comprehensive network intrusion detection dataset generated over 5G wireless network. IEEE Dataport. 2022. doi:10.21227/xtep-hv36.

23. Manasa K, Leo Joseph LMI. A real time dataset “IDSIoT2024” for machine learning/deep learning based cyber attack detection system for IoT architecture. In: 2025 3rd International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT). Bengaluru, India: IEEE; 2025. doi:10.21227/gfaz-t124.
24. Arjovsky M, Chintala S, Bottou L. Wasserstein generative adversarial networks. In: International Conference on Machine Learning. Sydney, Australia: PMLR; 2017. p. 214–23.