Applied Soft Computing Journal 168 (2025) 112455

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Anomaly-based network intrusion detection using denoising autoencoder and Wasserstein GAN synthetic attacks

Mohammad Arafah [a,*], Iain Phillips [b], Asma Adnane [b], Wael Hadi [a], Mohammad Alauthman [a], Abedal-Kareem Al-Banna [c]

[a] Department of Information Security, University of Petra, Amman, Airport Rd. 317, Jordan
[b] Department of Computer Science, Loughborough University, Loughborough, LE11 3TU, United Kingdom
[c] Department of Data Science and Artificial Intelligence, University of Petra, Amman, Airport Rd. 317, Jordan

ARTICLE INFO

Keywords:
AIDS
WGAN
NSL-KDD
CIC-IDS2017
Imbalanced dataset
Synthetic attacks

ABSTRACT

Intrusion detection systems face challenges in handling high-dimensional, large-scale, and imbalanced network traffic data. This paper proposes a novel architecture combining a denoising autoencoder (AE) and a Wasserstein generative adversarial network (WGAN) to address these issues. The AE-WGAN model extracts high-representative features and generates realistic synthetic attacks, effectively resolving data imbalance and enhancing anomaly-based intrusion detection. Extensive experiments on NSL-KDD and CICIDS-2017 datasets, using both binary and multiclass classification scenarios with various classifier architectures, demonstrate the model's superior performance. The proposed approach outperforms state-of-the-art models in accuracy, precision, recall, and F1 score, showing excellent generalization capabilities against unseen attacks. Time complexity analysis reveals computational efficiency while maintaining high-quality synthetic attack generation. This research contributes a robust, efficient, and adaptable framework for intrusion detection, capable of handling modern network traffic complexities and evolving cyber threats.

1. Introduction

The rapid growth of data globally leaves vast challenges to cybersecurity. Now, growing data come as a response to developing technologies and infrastructures, such as data transmission, communication systems, and the Internet of Things [1]. As a result, the number of attacks continuously increases over time, leading to financial losses, denial of services, and other adverse effects for countries and organizations [2].

From this point, detecting network intrusions with highly accurate performance in a short time can be a significant response as a defence strategy in cybersecurity. For this reason, an Intrusion Detection System (IDS) is extensively used to mitigate losses and damages caused by cyberattacks.

While previous works have explored GANs for synthetic attack generation, the combination of a denoising autoencoder with WGAN for intrusion detection remains unexplored. This research addresses the gap by proposing a novel AE-WGAN architecture that leverages the denoising capabilities of autoencoders to enhance the quality of synthetic attacks generated by WGAN, thereby improving the robustness of anomaly-based intrusion detection systems against both common and rare attack types.

IDS can be divided based on a monitoring scope or detection methodology [3]. From the monitoring scope perspective, it is divided into *Network-based Intrusion Detection Systems (NIDS)* and *Host-based Intrusion Detection Systems (HIDS)*, where the NIDS analyses and monitors the network traffic at the network level where the HIDS at the host level. In terms of detection methodology, IDS can be categorized into two main types: *Signature-based IDS (SIDS)* and *Anomaly-based IDS (AIDS)*, where SIDS depends on predefined rules, while AIDS depends on attack behaviour for the detection process. Consequently, attacks change their behaviours to appear like new attacks, which tricks SIDS performances compared to AIDS, which is built based on *Machine Learning (ML)* and *Deep Learning (DL)* approaches [4].

While previous works have explored various GAN architectures for intrusion detection, the novel combination of a denoising autoencoder with WGAN offers unique advantages. The denoising autoencoder serves as a powerful feature extractor, reducing noise and capturing essential characteristics of network traffic. This refined representation is then fed into the WGAN, enabling it to generate higher quality

* Corresponding author.
E-mail addresses: Mohammad.Arafah@uop.edu.jo (M. Arafah), i.w.phillips@lboro.ac.uk (I. Phillips), a.adnane@lboro.ac.uk (A. Adnane), whadi@uop.edu.jo (W. Hadi), mohammad.alauthman@uop.edu.jo (M. Alauthman), abanna@uop.edu.jo (A.-K. Al-Banna).

https://doi.org/10.1016/j.asoc.2024.112455
Received 8 May 2024; Received in revised form 9 October 2024; Accepted 5 November 2024
Available online 22 November 2024

synthetic attacks. Unlike existing approaches that use GANs directly on raw data, this method leverages the strengths of both autoencoders and GANs, resulting in more realistic and diverse synthetic samples. This synergy between AE and WGAN contributes to improved detection of both common and rare attack types, addressing a key challenge in intrusion detection systems.

SIDS relies on predefined rules to identify known attack patterns, whereas AIDS utilizes ML and DL techniques to analyse attack behaviours for detection. As a result, attackers often modify attacks behaviours' to evade SIDS, making AIDS more resilient due to its ability to adapt to new attack variations.

Further, ML-based AIDS performs poorly for complex and high-dimensional traffic data. Also, building AIDS based on an imbalanced dataset delivers limited performances for rare attacks (minor classes), where the ML classifier learns from major classes more than minor classes.

Recent advancements in AI-enabled network intrusion detection systems have shown promising results in addressing the challenges of high-dimensional data and evolving attack patterns. Kumar et al. provide a comprehensive survey of these techniques, highlighting the trend towards deep learning approaches and the increasing focus on handling imbalanced datasets. This aligns with the proposed AE-WGAN architecture, which aims to address these very challenges. Recent surveys on AI-enabled network intrusion detection systems [5] highlight the growing importance of machine learning in cybersecurity, providing a foundation for the AE-WGAN approach

As a result, DL has been employed in building and obtaining high-quality adversarial attacks for AIDS. A DL approach can learn from complex and high dimensional traffic data through deep neural network architecture with low error detection. Also, it has been employed to build generative adversarial networks like WGAN to mitigate an imbalanced dataset challenge [6].

There are three methods to mitigate imbalanced dataset impact on AIDS: Data Level, Cost-Sensitive, and Algorithms Level [7]. This research focuses on the data level, which includes two techniques: *Oversampling* and *Downsampling* . Oversampling means obtaining more samples for minor classes, while Downsampling reduces the number of samples for major classes. In this research, AE-WGAN belongs to the oversampling technique, which improves AIDS performance compared with traditional WGAN and state-of-the-art models.

To summarize, AIDS has superior attack detection against unseen attacks compared to SIDS. For generalization detection, the DL-based AIDS performs better than the ML-based one in terms of performance metrics. However, AIDS faces a challenging mission in detecting original attacks from synthetic ones. Besides, high dimensionality for large-scale traffic and imbalanced traffic data can significantly affect AIDS performance, including a detection rate metric. In order to overcome these challenges, this study proposes a novel model to build a robust AIDS that considers extracting complex features for high-dimensionality traffic data and generating high adversarial quality attacks to train AIDS. The main contributions of the proposed system in this paper are as follows:

- Extract high-level abstract features based on the *Analysis of Variance (ANOVA)* method, which can determine the relevant attack features. ANOVA applies statistics computations, which find the differences between features rather than the proposed complex models based on DL, which requires a large number of parameters to update in the training process.
- Design a new architecture that outperforms WGAN performance through denoising AE, which can learn precisely from network traffic for the large-scale, high-dimensional, complex network traffic features to obtain high detection scores with a low complexity model.
- Improve adversarial attacks' quality and stability via applying AE-WGAN architecture for minority and majority attacks compared with WGAN architecture and current approaches.
- Build a robust AIDS based on adversarial attacks of AE-WGAN architecture against unseen attacks under official splits of NSL-KDD and CICIDS-2017 datasets.
- Evaluate AIDS-AE-WGAN based on different DL classifiers and types of classification without bias towards a specific model.

2. Related works

With the development of DL techniques, several models have been proposed for AIDS to extract features from large-scale, high-dimensional, and complex traffic data [7]. Generative adversarial networks have been constructed based on DL architecture to resolve imbalanced data issues.

Ieracitano et al. [8] proposed a hybrid model combining AE and statistical techniques for AIDS. Their approach outperformed traditional ML and DL-based AIDSes, achieving F1 scores of 81.98% and 82.04% for binary and multi-classification on the NSL-KDD dataset, respectively.

Xu et al. [3] introduced the Log-cosh Conditional Variational AutoEncoder (LCVAE) to tackle imbalanced datasets. Their model, applied to the NSL-KDD dataset, achieved 85.51% accuracy and 80.78% F1 score, demonstrating improved performance in handling complex data.

Lee and Park [9] employed AE with conditional GANs to improve AIDS performance on the CICIDS-2017 dataset. Their approach showed high performance in multiclass classification, particularly for rare attacks.

Oguz Kaplan and Emre Alptekin [10] proposed a one-class anomaly detection algorithm using BiGAN architecture to mitigate imbalanced data challenges. Their model performed well on the KDDCUP99 dataset but did not include multi-classification for generalization detection.

Gulrajani et al. [11] proposed an improved WGAN model with gradient penalty, achieving better training stability. Their approach outperformed the original WGAN architecture on various datasets.

Yun et al. [12] introduced the HMCD-Model for detecting HTTP-based malicious communication traffic. Their model, using WGAN-GP and a hybrid CNN-LSTM approach, showed efficient performance on the CIC-IDS2017 dataset.

Zhang et al. [7] designed the CWGAN-CSSAE model to improve AIDS performance for unseen attacks and resolve imbalanced dataset challenges. Their approach achieved high F1 scores on KDDTest-21 and UNSW-NB15 datasets, particularly for minority attack detection.

Recent research has explored federated learning for intrusion detection. Li et al. [13] proposed FedIDS, a federated framework for 5G networks, while Liu et al. [14] applied federated learning to IoT network anomaly detection.

Cui et al. [15] proposed the GMM-WGAN-IDS model, combining feature extraction, imbalance processing, and classification modules. Their approach showed improved performance on NSL-KDD and UNSW-NB15 datasets compared to existing models.

The models mentioned above applied various combinations of AE and generative adversarial networks to improve attack detection performance. However, little attention has been paid to utilizing AE alongside WGAN capabilities, where WGAN offers more stable training than traditional GANs. This research gap motivates the proposed AE-WGAN approach, which aims to significantly improve the quality of generated attacks and overall detection performance.

Recent work by Gide and Mu'azu [16] has demonstrated the effectiveness of hybrid approaches in real-time intrusion detection for IoT networks. Their model, combining KNN and dense neural networks, achieved high accuracy in classifying (DoS/DDoS) attacks on the MQTT-IoT-IDS2020 dataset, with an AUC score of 95.7%. This study aligns with the AIDS-based AE-WGAN model, highlighting the potential of hybrid models in enhancing IoT security and emphasizing the importance of real-time detection capabilities for emerging attack patterns [16].

Table 1 summarizes key approaches in recent literature, highlighting the novelty of the AE-WGAN method in addressing both common and rare attack detection across multiple datasets.

Table 1
Comparison of different methods for attack detection

Study	Method	Dataset	Key findings
Zhang et al. [7]	CWGAN-CS	NSL-KDD	Improved minority attack detection
Lee and Park [9]	AE-CGAN	CICIDS-2017	Enhanced performance for rare attacks
Cui et al. [15]	GMM-WGAN-IDS	NSL-KDD, UNSW-NB15	Better handling of imbalanced data
Yun et al. [12]	HMCD-Model	CICIDS-2017	Effective for HTTP-based malicious traffic
Our approach	AE-WGAN	NSL-KDD, CICIDS-2017	Superior performance for common and rare attacks

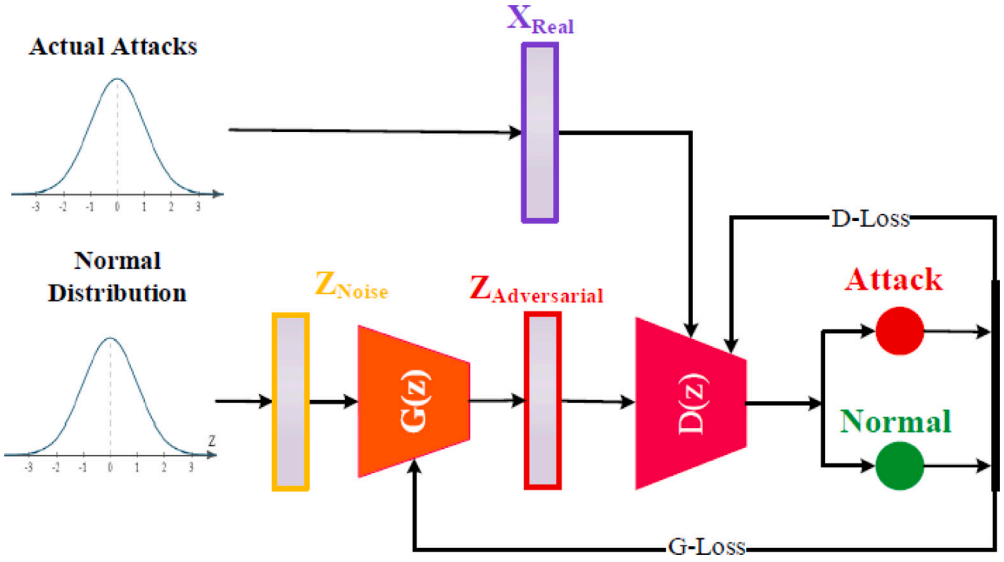


Fig. 1. GAN's architecture.

3. Background

3.1. Fundamentals of GAN's

GAN is a generative model used in different disciplines [17]. In cybersecurity, it is used to mitigate an imbalanced dataset to obtain powerful AIDS against unseen and rare attacks. It consists of two neural networks; a generator (G) and a discriminator (D). The G generates adversarial attacks based on an input vector (z), while the D discriminates between the actual attack (x) and adversarial attack $G(z)$. Goodfellow et al. proposed that G and D work in an adversarial environment, such as a zero-sum game. The Eq. (1) shows the GAN's objective and cost for each component [18].

$$\min_{\theta G}\max_{\theta D}V(G,D) = \min_{G}\max_{D}E_{x\sim \text{Pdata}}[\log D(x)] + E_{z\sim \text{Pz}}[\log(1 - D(G(z)] \tag{1}$$

The GAN's objective is to mimic an actual data distribution P_G into adversarial distribution P_z with high convergence to P_G. In other words, G tries to generate samples that a D cannot distinguish from actual attacks. The optimal situation for the G is when the D gives a 0.5 classification probability between adversarial and actual attack, while the D tries to defeat the G by classifying adversarial attacks correctly. However, G and D enhance their performance over a training process, where the loss error is reduced in a backpropagation process for each one. Fig. 1 shows the GAN's components (G, D) in the context of an adversarial environment.

GAN's model is an unsupervised learning type, which means no label should exist for network traffic to measure performance. Although GAN's model can produce high-quality attacks, the ideal case of GAN on a real dataset rarely occurs. The input sample, learning process, and the used architecture can affect attack quality where the GAN is a probabilistic model and works under binary cross-entropy, as explained in Eq. (1).

3.2. Wasserstein Generative Adversarial Network (WGAN)

GAN faces several issues, like mode collapse (gradient disappearance), stability, and non-convergence between original and adversarial distributions. Thus, Arjovsky et al. proposed WGAN to resolve upper mentioned issues. As a result, a learning curve and optimization process become more smooth and have less computation cost [19]. The main difference between GANs and WGAN is distinguished by measuring the difference between two distributions (p_r, p_g).

WGAN uses Wasserstein's (Earth Mover's) distance instead of Jensen Shannon Divergence (JSD), making this type more stable than the standard GAN. Eq. (2) shows the differences between p_r and p_g distributions by finding the best plate $\Pi(p_r, p_g)$ using the lowest term inf for x, which is mapped to y. The best plate means a lower number of computations, so the joint distribution was used to achieve this objective.

$$W(p_r, p_g) = \inf_{\gamma\sim\Pi(p_r,p_g)} E_{(x,y)\sim\gamma}\|x - y\| \tag{2}$$

The reason behind using Wasserstein's distance instead of JSD and KL divergence is to avoid the overlaps between p_r and p_g distributions. When the theta value equals zero, the KL and JSD values become infinity because these functions are not differentiable. In contrast, a Wasserstein's distance becomes zero, as shown in Eqs. (3).

$$\begin{aligned} D_{KL}(P \parallel Q) &= \sum_{x=0,y\sim(0,1)} 1\cdot\log\frac{1}{0} = +\infty \\ D_{KL}(Q \parallel P) &= \sum_{x=0,y\sim(0,1)} 1\cdot\log\frac{1}{0} = +\infty \\ D_{JS}(P,Q) &= \frac{1}{2}\Big(\sum_{x=0,y\sim(0,1)} 1\cdot\log\frac{1}{\frac{1}{2}} + \sum_{x=0,y\sim(0,1)} 1\cdot\log\frac{1}{\frac{1}{2}}\Big) = \log 2 \\ W(P,Q) &= |\theta| \end{aligned} \tag{3}$$

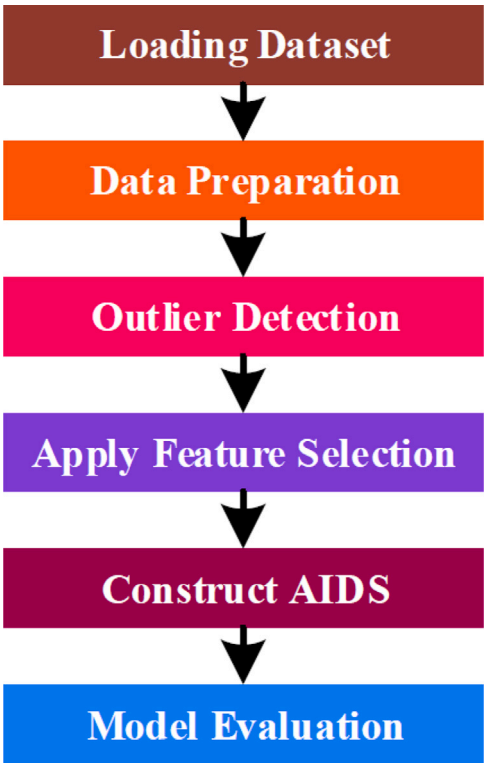


Fig. 2. Framework's modules.

4. Proposed model

4.1. Framework

The framework proposed for AIDS based on AE-WGAN includes four major modules: data preprocessing, data oversampling, building AIDS, and evaluation results, as indicated in Fig. 2.

4.2. Data preprocessing

Generative adversarial models are sensitive to data input since it depends on different parameters. Thus, data preprocessing is essential for delivering high adversarial attack quality. Therefore, data cleaning, encoding, scaling, and feature selection were applied in this research, as indicated in Fig. 3.

Data cleaning was applied to remove network traffic features that contain fixed values over the entire dataset (constant features) and for features that mostly hold constant values (quasi-constant features) followed by an ordinal encoder. The ordinal encoder was used to convert all non-numeric features into numeric values. After that, numeric features were scaled through the min–max function, which uniforms all features into the same scale. Consequently, the learning process time was faster since the amount and number of changes were decreased in the training process. Feature selection was conducted after applying feature scaling using ANOVA feature selection to obtain high-quality attacks. ANOVA is a statistical method applied to find the correlation features based on variances between independent features (variables) that lead to predicting the label class precisely.

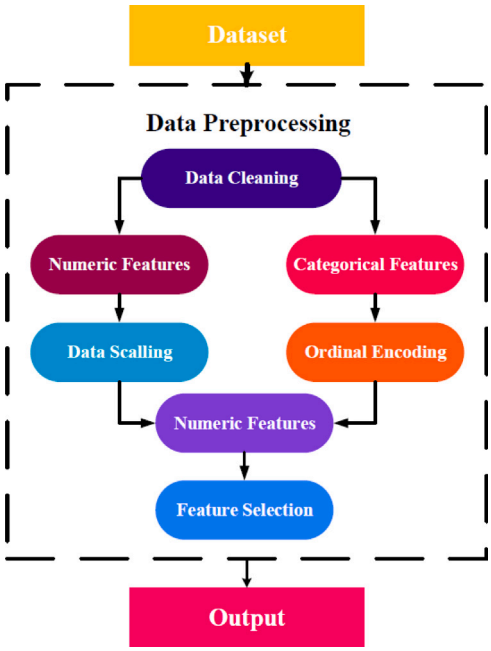


Fig. 3. Data preprocessing steps.

4.3. Data oversampling

4.3.1. Denoising AE

This module produces adversarial attacks based on AE-WGAN architecture, which consists of AE and WGAN. AE can enhance attack quality through a representation of attacks. From an architecture perspective, AE includes two components: Encoder (E) and Decoder (D) since the E converts a high dimensionality input vector into a lower representation (latent) vector, where the D the encoded vector and reconstructs a similar input vector [20].

In this research, AE is employed in WGAN to utilize the quality of generated attacks by reducing the noise in the WGAN model. The noises are reduced by E, which takes the input x to map it into output h through the mapping function, as mentioned in Eq. (4).

$$h = f(Wx + b) \tag{4}$$

However, the D function performs the reconstruction process, which takes the latent vector (h) to build an x vector, as shown in Eq. (5). The f' in Eq. (5), represents the decoder function, which updates the weights of neurons (w') input and adds it to the bias (b') constant value.

$$x' = f'(W'h + b') \tag{5}$$

The reconstructed vector obtained by E through the mapping function decreases the classification error for a discriminator in WGAN. As a result, the adversarial environment in WGAN enhances attack quality by updating the loss generator in backpropagation. The mapping function in E is non-linear since the AE learns from an attack representation deeply. Fig. 4 shows the AE architecture used in WGAN.

4.3.2. AE-WGAN

In this research, AE considers an essential module employed in WGAN architecture to obtain high-quality attacks. To resolve the mentioned challenges, AE-WGAN architecture is built to achieve this objective. Fig. 5 shows the proposed architecture, which is outperformed in performance compared with a traditional WGAN model.

The AE component consists of an encoder with 3 dense layers (24-20-16 neurons) and a decoder with 3 dense layers (16-20-24 neurons), both using ReLU activation. The WGAN generator uses 3 dense layers (32-64-24 neurons) with ReLU activation, while the critic uses 3 dense layers (24-16-1 neurons) with linear output. The study utilized the Adam optimizer with a learning rate of 1e−4 for both AE and WGAN, training for 100 epochs with a batch size of 32. The WGAN critic was updated 5 times per generator update to ensure stability.

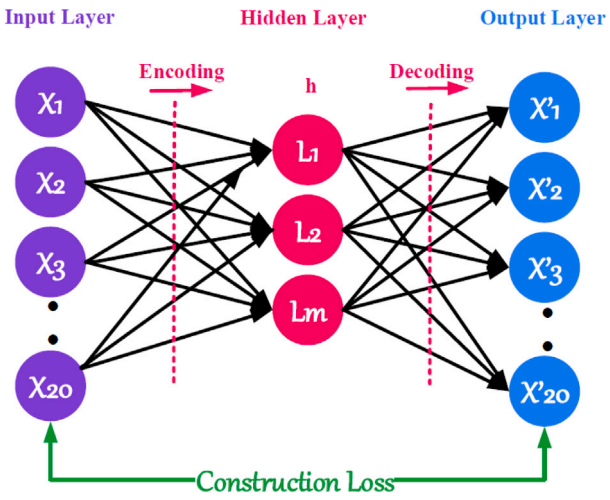


Fig. 4. AE architecture.

The AE-WGAN architecture starts by passing a vector from processed network traffic into the AE architecture. The length of the input vector to AE-WGAN architecture is 24, where it is encoded as a latent vector with a lower length of 20 without noise. After that, the latent vector is constructed with the same input for the encoder ($length = 24$) with deep representation learning as an output for the decoding process. ReLU activation function is applied in AE architecture for the encoding process.

WGAN in the proposed architecture (AE-WGN) starts with a random noise vector (z) from a Gaussian distribution with a length of 32. The z vector is passed into the generator network to obtain a new adversarial attack. The generator network consists of three layers, where a ReLU activation function is applied in all layers, with (40, 80, and 120) for a dense parameter, respectively. In the training process of AE-WGAN architecture, the generator is trained independently to produce adversarial attacks close to the real ones. However, the generation errors in the generator are corrected by updating the neurons' weights in a backpropagation function. Consequently, the generator produces high adversarial attacks quality.

Once the generator is trained, the discriminator training starts to discriminate between the original and adversarial attacks. Due to its functionality, the discriminator network is easier than the generator network architecture. Thus, the dense discriminator parameter for the layers are (80, 40, and 1) with a ReLU activation function for the input, hidden, and output layers, respectively. Now, the encoded vector and z noise are passed to the discriminator. The discriminator is trained to discriminate between the encoded vector (real attack) and the adversarial attack produced by the generator. The discriminator tries to discriminate between them with low classification errors. Due to the Wasserstein distance used in WGAN architecture, a critic function is used instead of a discriminator.

AE-WGAN architecture used the Adamax algorithm for optimization, where the learning rate was 0.00009. The proposed architecture was trained with 101 epochs, and the batch size was 32, which enhanced the stability compared with traditional WGAN to be considered a general solution.

The AE-WGAN architecture is trained several times based on a set of steps for each network: generator and discriminator. The complete steps are indicated in Algorithm 1.

The time complexity of the AE-WGAN model is O(nme), where n is the number of samples, m is the number of features, and e is the number of epochs. The autoencoder contributes O(nm) for encoding and decoding, while the WGAN adds O(ne) for generator and discriminator updates. This makes the approach computationally efficient compared to more complex deep learning architectures, while still providing high-quality synthetic attacks.

Algorithm 1 An algorithm AE-WGAN for obtaining adversarial attacks

Input: Take X_i samples from an AIDS dataset (NSL-KDD or CICIDS-2017).

Output: Produce adversarial attacks by applying $AE - WGAN(N)$.

1: Take N vector from Gaussian distribution.
2: Apply generator G on N sample: $G(N) \implies P_G$ adversarial.
3: Apply AE on real network traffic x_i: $AE(x_i) \implies P_r$ (real distribution).
4: Apply backpropagation to generator only as follows: $g_\theta \leftarrow -\nabla_\theta \frac{1}{m} \sum_{i=1}^{m} f_w\left(g_\theta\left(z^{(i)}\right)\right)$.
5: Repeat **Step 4** until generator produce close to real attack r.
6: Train the discriminator for real and normal samples only by applying critic function.
7: Repeat **Step 5** until the discriminator produce high detection accuracy.
8: Train both networks generator and discriminator D($x_i \sim P_r$, $N_i \sim P_G$).
9: Repeat **Step 8** until the highest convergence between both distributions obtained (P_G, P_r).

Table 2
NSL-KDD counters over training and testing sets.

NSL-KDD	Normal	DoS	Probe	R2L	U2R	Total
Training set	67,343	45,927	11,656	995	52	125,973
Testing set	9710	5741	1106	2199	200	18,956

The implementation of the proposed AE-WGAN model is available on GitHub [21], allowing for reproducibility and further experimentation by the research community.

4.4. Building AIDS

4.4.1. NSL-KDD dataset

NSL-KDD is a network-based IDSs made by the Canadian Institute of Cybersecurity [13]. It is a subset of the KDD'99 dataset, which avoids redundant records and the limitation of the KDD'99 dataset. Therefore, AIDSes are built and compared based on the NSL-KDD since it is a clean and realistic dataset. NSL-KDD was split into two subsets: training and testing. For realistic evaluation, only seventeen attack types existed in the testing set, which made the evaluation reliable. Table 2 shows the attacks' counters over categories in each split.

It is important to note that while the NSL-KDD dataset addresses some limitations of the original KDD'99 dataset by removing redundant records, it still presents challenges with class imbalance. The AE-WGAN approach not only helps in generating synthetic samples for underrepresented classes but also aids in creating diverse variations of existing attack patterns. This diversity is crucial for improving the model's ability to detect subtle variations of known attacks and potentially identify novel attack patterns. By carefully balancing the augmented dataset, the approach ensures that the model does not overfit to the majority classes while still maintaining sensitivity to rare but potentially critical attack types like U2R and R2L.

To address the class imbalance in the NSL-KDD dataset, particularly for the underrepresented U2R and R2L attack classes, the study employed the AE-WGAN architecture to generate synthetic samples. For each minority class, synthetic samples were generated until their representation reached 10% of the majority class (Normal). This approach helps to mitigate the tendency of machine learning algorithms to ignore less frequent attack types, which is crucial for detecting potentially severe U2R and R2L attacks. The synthetic samples were carefully validated to ensure they maintained the statistical properties of the

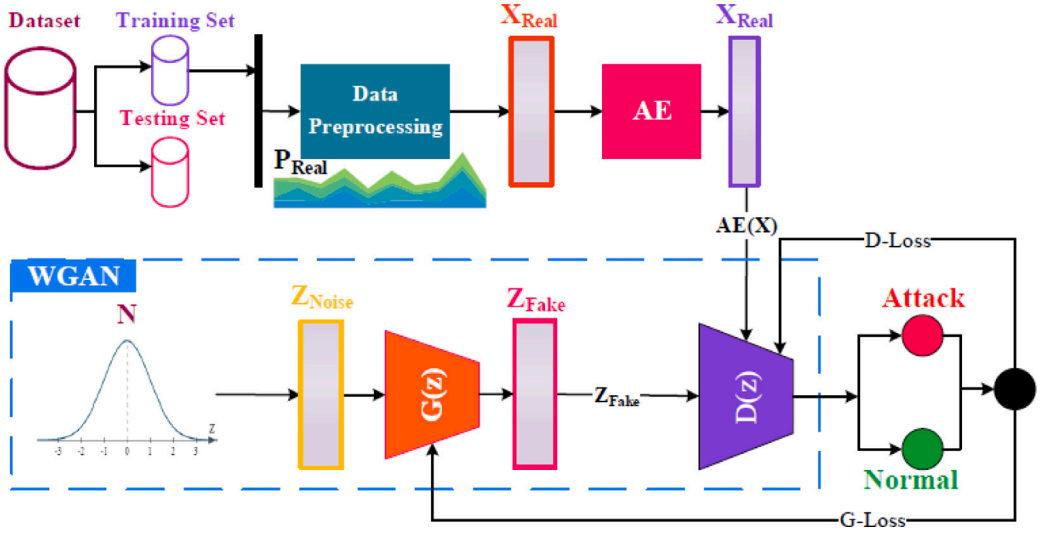


Fig. 5. AE-WGAN architecture.

Table 3
Attacks collected in the CICIDS-2017 dataset over five days.

Days	Classes
Monday	Benign
Tuesday	Brute Force, FTP-Patator, SSH-Patator
Wednesday	DoS/DDoS, DoS slowloris, DoS Slowhttptest, DoS Hulk, DoS GoldenEye, Heartbleed
Thursday	HeartBrute Force, XSS, SQL Injection, Infiltration
Friday	DDoS LOIT, Botnet ARES, PortScans

original attacks while introducing sufficient variability to improve model generalization.

The NSL-KDD is characterized by reasonable distribution for the number of samples in each set. As a result, the random selection of samples is unnecessary for building AIDS. Also, it is a free and public dataset that researchers extensively use. However, the NSL-KDD dataset considers an imbalanced dataset, where the AE-WGAN architecture is applied to obtain more attacks for the majority and minority classes in this research.

4.4.2. CICIDS-2017

The CICIDS-2017 is an Anomaly Intrusion Detection (AID) dataset that covers different datasets' limitations. The limitations are due to the lack of traffic, features, attacks' diversity and anonymous payload data [22]. The traffic data in CICIDS-2017 represents real traffic in packet capture format. Therefore, human behaviours were collected by the B-Profile system for 25 users in a dedicated network. The network traffic was collected over five days, while different protocols were involved (SMTP, HTTP, HTTPS, FTP, and SSH). Each day, specific attacks and benign records are collected at specific times since this traffic has complete interaction and capture. In addition, the total number of flow features was 80, which were extracted using a network traffic flow generator and analyser known as a CICFlowMeter. Table 3 shows the attacks' names over five days.

To handle the large scale and diversity of the CICIDS-2017 dataset, a stratified sampling approach was employed to ensure representation of all attack types in both training and testing sets. For extremely rare attacks like Heartbleed, additional synthetic samples were generated using the AE-WGAN model to ensure sufficient representation for training. The dataset was then normalized using min–max scaling to ensure all features were on a comparable scale, which is particularly important for the diverse set of features in CICIDS-2017.

4.5. AIDS architecture

To build a robust AIDS, AE-WGAN architecture is used to obtain high-quality attacks through a training process. Since different classifiers can deliver different results in detecting anomaly attacks, several classifiers are constructed and trained based on adversarial attacks produced by AE-WGAN architecture. As a result, Decision Tree (DT), Random Forest (RF) are applied for the ML approach, while Convolutional Neural Network (CNN), Gated Recurrent Unit (GRU), Long Short Term Memory (LSTM), and Recurrent Neural Network (RNN) for DL approach.

All upper-mentioned classifies are constructed and tuned to deliver high attack detection performance based on AE-WGAN architecture. Therefore, unbiased towards a specific classifier is ensured, which confirms that the proposed architecture considers a general solution where NSL-KDD and CICIDS-2017 datasets are employed. In addition, two types of classification are conducted to show the difficulties against AIDS based on AE-WGAN architecture. Table 4 shows the description of architectures used in building AIDSes.

5. Experiment results

5.1. Evaluation metrics and environmental configuration

5.1.1. Evaluation metrics

In this research, AE-WGAN AIDS-based is evaluated based on Accuracy, Precision, Recall, and F1 score, which are extensively used in related works. All performance metrics are derived from a confusion matrix. It contains four values: True positive (TP), False Negative (FN), False Positive (FP), and True Negative (TN). The TP represents the samples predicted as attacks and labelled with actual attacks. The FN is like TP, but it is predicted as normal by a model. The FP represents the samples predicted as attacks while labelled with actual

Table 4
Description of architectures used based on AE-WGAN.

Classifier	NSL-KDD		CICIDS-2017	
	Binary	Multiclass	Binary	Multiclass
DT	Max depth = 7	Max depth = 9	Max depth = 3	Max depth = 10
RF	Max depth = 8	Max depth = 10	Max depth = 9	Max depth = 7
CNN	Input Layer: Conv2D(40), Kernel size=(3, 3), AF = ReLU. Dropout(0.1) Hidden Layer: Dense(40), AF = ReLU Output Layer: Dense(1), AF = sigmoid	Input Layer: Conv2D(40), Kernel size=(3, 3), AF =ReLU. Dropout(0.1) Hidden Layer: Dense(40), AF = ReLU Output Layer: Dense(5), AF = sigmoid	Input Layer: Conv2D(40), Kernel size=(1, 1), AF =tansh Hidden Layer: Dense(40), AF = tansh Output Layer: Dense(1), AF = sigmoid	Input Layer: Conv2D(40), Kernel size=(1, 1), AF =tansh Hidden Layer: Conv2D =40, Kernel size = (1,1), AF = tansh. Dense(40), AF = tansh Output Layer: Dense(15), AF = softmax
GRU	Input Layer: GRU(30), AF = ReLU. Dropout(0.1) Hidden Layers: Dense(10), AF= ReLU Dense(10), AF = ReLU Output Layer: Dense(1), AF = sigmoid.	Input Layer: GRU(30), Return sequences=True, AF = ReLU. Dropout(0.1) Hidden Layers: GRU(30), AF = ReLU. Dropout(0.1) Dense(30), AF = ReLU. Dropout(0.1) Dense(30), AF = ReLU. Output Layer: Dense(5), AF = softmax	Input Layer: GRU(40), Return sequences=True, input shape (2,5), AF = tansh Hidden Layers: GRU(40), AF = tansh, Return sequences = True. GRU(40), AF = tansh. Dense = 40, AF = tansh, kernel regularizer l1(0.005) Output Layer: Dense = 1, AF = sigmoid	Input Layer: GRU(40), Return sequences=True, input shape (2,5), AF = tansh. Hidden Layers: GRU(40), AF = tansh, Return sequences = True. GRU(40), AF = tansh. Dense(40), AF = tansh. Output Layer: Dense(15), AF = softmax
LSTM	Input Layer: LSTM(10), AF = ReLU. Hidden Layers: LSTM(20), AF = ReLU. Dropout(0.5) Dense(10), AF = ReLU Output Layer: Dense(1), AF = sigmoid	Input Layer: LSTM(10), AF = ReLU. Dropout(0.1) Hidden Layers: LSTM(20), AF = ReLU. Dropout(0.2) LSTM(20), AF = ReLU. Dropout(0.1) Dense(10), AF = ReLU Output Layer: Dense(5), AF = softmax	Input Layer: LSTM(40), input shape(2,5), AF = tansh, Return sequences = True. Hidden Layers: LSTM(40), input shape(2,5), AF = tansh, Return sequences = True. LSTM(40), input shape(2,5), AF = tansh. Dense(40), AF = tansh. Output Layer: Dense(1), AF = sigmoid	Input Layer: LSTM(40), input shape(2,5), AF = tansh, Return sequences = True. Hidden Layers: LSTM(40), input shape(2,5), AF = tansh, Return sequences = True. LSTM(40), input shape(2,5), AF = tansh. Dense(40), AF = tansh. Output Layer: Dense(15), AF = softmax
RNN	Input Layer: SimpleRNN(30), AF = ReLU. Dropout(0.1) Hidden Layers: Dense(10), AF = ReLU Dense(10), AF = ReLU Output Layer: Dense(1), AF = sigmoid	Input Layer: SimpleRNN(50), AF = ReLU. Dropout(0.2) Hidden Layers: SimpleRNN(50), AF = ReLU. Dropout(0.2) SimpleRNN(50), AF = ReLU. Dropout(0.2) SimpleRNN(50), AF = ReLU. Dropout(0.2) Output Layer: Dense(5), AF = softmax	Input Layer: SimpleRNN(40), input shape(2,5), AF = tansh, Return sequences = True. Hidden Layers: SimpleRNN(40), input shape(2,5), AF = tansh, return sequences = True. SimpleRNN(40), input shape(2,5), AF = tansh Output Layer: Dense(1), AF = sigmoid	Input Layer: SimpleRNN(40), input shape(2,5), AF = tansh, Return sequences = True. Hidden Layers: SimpleRNN(40), input shape(2,5), AF = tansh, return sequences = True. SimpleRNN(40), input shape(2,5), AF = tansh Output Layer: Dense(15), AF = softmax

normal. TN represents the number of normal samples labelled with actual normal and predicted as normal.

Accuracy is the proportion of samples detected as benign or attacks correctly, as shown in Eq. (6).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

Precision or Positive Predicted Value (PPV) is the proportion of samples detected as attacks over all samples predicted by the model, as shown in Eq. (7).

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

Recall or Detection Rate (DR) is the portion of samples detected as attacks' overall actual attacks' labels, as shown in Eq. (8).

$$Recall = \frac{TP}{TP + FN} \tag{8}$$

F1 score is a harmonic function (mean) that produces a range value (0,1). The value reflects the mean of precision and recall, where a higher value means better attack detection, as shown in Eq. (9).

$$F - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{9}$$

The F1 score is particularly significant for evaluating models on imbalanced datasets, as it provides a balanced measure of both precision and recall. It was chosen as the primary metric for improvement calculation because it offers a single, comprehensive measure of a model's performance, especially important when dealing with rare attack types in intrusion detection systems.

Recall and F1 score are suitable metrics to evaluate AIDS performance for imbalanced datasets. For the generative adversarial models, cosine similarity is used to measure the distance between original and generated distributions, as shown in Eq. (10).

$$\cos(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x}.\mathbf{y}}{\| \mathbf{x} \| \times \| \mathbf{y} \|} \tag{10}$$

5.1.2. Environmental configuration

The environment configuration was built to conduct all experiments to show the impact of AE-WGAN on AIDSes performance. So, the Integrated Development Environment (IDE) used for code development was Spyder, which runs on Windows 10. For programming language, Python 3.6 was used to conduct all experiments, where the standard packages were employed, such as numpy, pandas, matplotlib, and scikit-learn. The hardware specification used was a ThinkPad $E15$ machine (Intel(R) Core(TM) i7-10510U CPU@1.80 GHz), where the AMD Radeon (TM) RX 640 GPU was used to accelerate the training and attacks detection.

5.2. Feature selection

Selecting relevant features of network traffic is essential to deliver high attack detection. So, Pearsons, Spearman, Kendall and ANOVA algorithms are investigated as feature selection methods on different datasets before obtaining adversarial attacks. The selected features from the upper-mentioned algorithms are evaluated in AIDS performance to show the best features that can be used for obtaining adversarial attacks in AE-WGAN architecture. The highest performance was obtained by ANOVA feature selection, which produced 91.0 % and 93.0 % F1 scores for NSL-KDD and CICIDS-2017, respectively. Table 5 shows the selected features for each step in the data preprocessing module.

Table 5
Feature selection module.

Preprocessing method	Dataset	Feature name	Number of features
Constant Features	NSL-KDD	num outbound cmds	1
Quasi-constant	NSL-KDD	Urgent, Num shells, Num access files, Is host login, Num failed logins, Duration, Hot, Num file creations, Is guest login, Service	10
Numeric Column	NSL-KDD	Duration, Source bytes, Destination bytes, Wrong fragment, Urgent, Hot, Number failed logins, Num compromised, Su attempted, Num root, Num file creations, Num shells, Num access files, Count, Srv count, Dst host count, Dst host srv count, Level	18
Feature Selection (ANOVA)	NSL-KDD	Protocol type, Flag, Land, Wrong fragment, Logged in, Count, Srv_count, Serror rate, Srv serror rate, Rerror rate, Srv rerror rate, Same srv rate, Diff srv rate, Srv diff host rate, Dst host count, Dst host srv count, Dst host same srv rate, Dst host diff srv rate, Dst host same src port rate, Dst host srv diff host rate, Dst host serror rate, Dst host srv serror rate, Dst host srv terror rate, Level	24
Constant Features	CICIDS-2017	BwdPSHFlags, FwdURGFlags, BwdURGFlags, RSTFlagCount, CWEFlagCount, ECEFlagCount, FwdAvgBytes/Bulk, FwdAvgPackets/Bulk, FwdAvgBulkRate, BwdAvgBytes/Bulk, BwdAvgPackets/Bulk, BwdAvgBulkRate	12
Feature Selection (ANOVA)	CICIDS-2017	DestinationPort, FlowDuration, BwdPacketLengthMin, FlowIATMean, FlowIATMax, FwdIATMean, FwdPackets/s, BwdPackets/s, MinPacketLength, min_seg_size_forward	10

5.3. Adversarial model performance

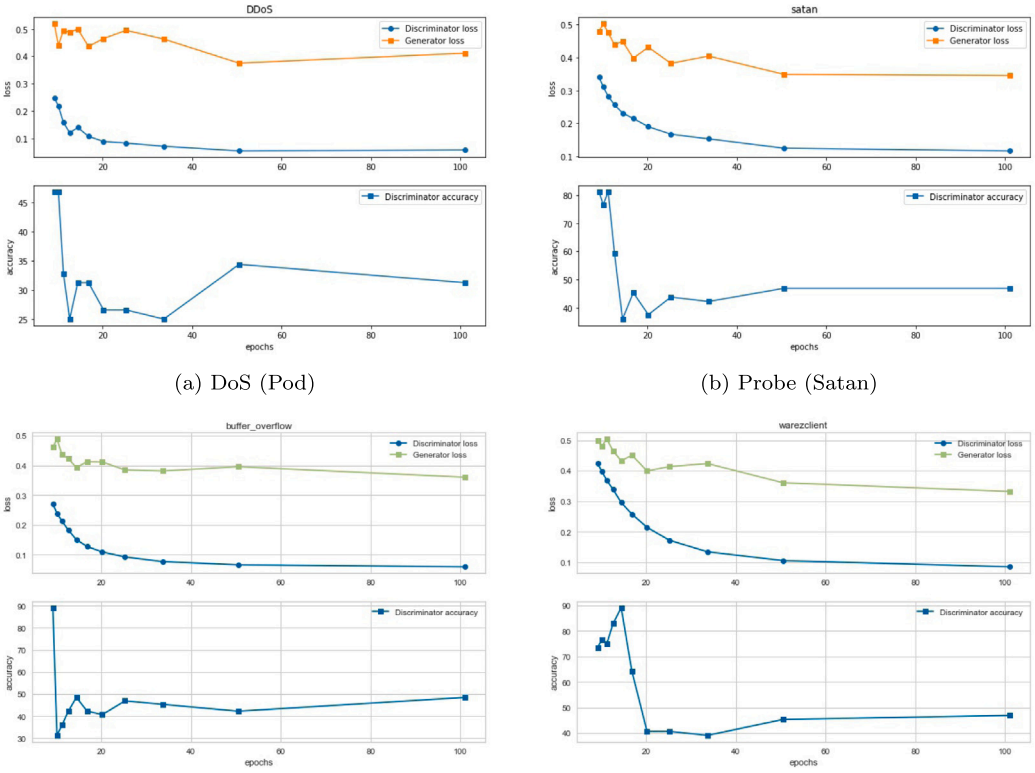
5.3.1. AE-WGAN training

The AE-WGAN architecture was trained to produce adversarial attacks based on the features selected by ANOVA. Since AE-WGAN is executed separately on each dataset, the adversarial attacks produced were added to the original set in each dataset to train an AIDS. According to the official split in NSL-KDD dataset, the total number of attack classes was 22 for training and 37 for testing sets. So, 22 attack types were used to train AE-WGAN architecture where the DOS, Probe, R2L, and U2R categories contained (six, four, seven, and five) attack types, respectively. The accuracy and loss error metrics over epochs were observed in the training process for each attack type to deliver high-quality attacks. Fig. 6 shows the AE-WGAN performance in the training process for different attacks in the DoS, Probe, R2L, and U2R categories.

AE-WGAN showed that the architecture was learned perfectly on the DoS (pod) attack since the number of samples was 201 compared to 54 for the U2R (Buffer overflow) attack, which produces a lower quality. In addition, the AE-WGAN was learned from Probe (Satan) better than the R2L(Warezclient) attack, where the number of samples was 3,630 and 890 for Probe (Satan) and R2L(Warezclient) attacks, respectively. In the training process for different types of attacks from different categories, stability was achieved even with attacks limited samples. Also, the generated attacks for Probe (Satan) and R2L(Warezclient) attacks were high-quality since the discriminator was unable to distinguish between original and adversarial attacks with 50.0 % accuracy.

Moving to the CICIDS-2017 dataset, AE-WGAN was performed perfectly for some attacks, such as DDoS and Bot, compared to Infiltration and Heartbleed attacks. The reason behind that is the number of samples, which was 128,027 and 1,966 for DDoS and Bot attacks, respectively.

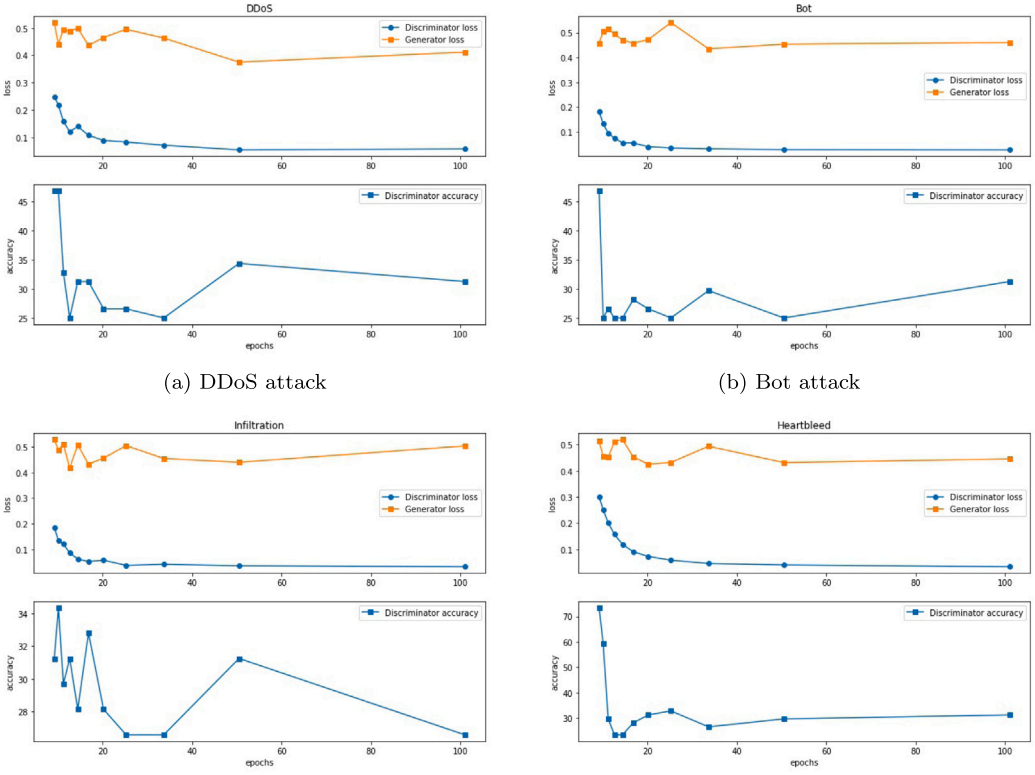
However, the AE-WGAN architecture produced adversarial attacks even with a limited number of samples, except for rare attacks. The results showed that the AE-WGAN could learn from the Infiltration attack with 36 samples but was unable to learn from the Heartbleed with 11 samples. As a result, the rare attack samples negatively affect the learning and the quality of produced attacks. Fig. 7 shows AE-WGAN training performance for the majority and minority attack classes.



(a) DoS (Pod) (b) Probe (Satan)

(c) R2L (Warezclient) (d) U2R (Buffer overflow)

Fig. 6. AE-WGAN training performance over different NSL-KDD categories.



(a) DDoS attack (b) Bot attack

(c) Infiltration attack (d) Heartbleed

Fig. 7. AE-WGAN architecture performance in CICIDS-2017.

5.3.2. AE-WGAN testing

The AE-WGAN architecture was applied to produce adversarial attacks based on the features selected by ANOVA. Since AE-WGAN was executed separately on each dataset, the adversarial attacks produced are used with the original set in each dataset to train an AIDS. Thus, the total number of samples used was 70,343, which balanced the number of samples in each attack category for both datasets, where the cosine similarity metric to the difference between original and adversarial distributions was computed for adversarial attacks. A high value in the cosine similarity means more similarity (high quality) to the original distribution traffic, while a lower value means the difference is large (lower quality).

The average cosine similarity for NSL-KDD and CICIDS-2017 datasets was 13.75 and 1.81, respectively. For more investigation, binary classification is conducted on adversarial and original attacks, where the original attacks were labelled as ones, and the adversarial attacks were labelled as zeros. Therefore a classifier was used to discriminate them. The results showed that the accuracy for CICIDS-2017 was 30 %, which means that the classifier could not often discriminate between adversarial and original attacks, compared with 74 % for the NSL-KDD.

The reason behind different performances for AE-WGAN architecture on two datasets returned to the attack's samples. The NSL-KDD contained 38, while the CICIDS-2017 contained 13 attacks. The total number of attack types less than 50 records was 18 in NSL-KDD, while CICIDS-2017 had only three attacks. Fig. 8 shows the distribution of NSL-KDD 8(a) and CICIDS-2017 (b) before and after applying the AE-WGAN architecture.

5.4. AIDS performance

5.4.1. Model performance on NSL-KDD

The evaluation of AIDS based on AE-WGAN architecture on the NSL-KDD dataset was initiated according to the type of classification. So, binary classification experiments were conducted on DT, RF, RNN, LSTM, GRU, and CNN classifiers. Fig. 9 shows the AIDS performance in terms of Accuracy, Precision, Recall, and F1 score.

According to Fig. 9, the DT classifier outperformed in terms of the F1 score, which reached 94 %, while GRU and RF were (5.0 %, 1.0 %) lower than the DT classifier, respectively. The LSTM classifier achieved 90 % recall, whereas the DT delivered 92 %. However, the RN, LSTM, and RF classifiers outperformed the DT and CNN, where the precision reached 97 %, while the DT and CNN were 96 % and 95 %, respectively. The accuracy performance was variant for the mentioned classifiers since LSTM, DT, and RF delivered the highest value compared to CNN, RNN, and GRU classifiers.

The superior performance of Decision Trees in binary classification, but not in multi-class classification, can be attributed to their inherent structure. In binary problems, DTs can create clear decision boundaries between two classes. However, as the number of classes increases, the tree's ability to partition the feature space efficiently decreases, leading to potential overfitting [23]. This phenomenon is well-documented in machine learning literature [24].

Since the model focused on producing high-quality attacks, AE-WGAN were evaluated over different attack categories to reveal the detection efficiency. Therefore, multi-class classification was conducted on the mentioned classifiers. Fig. 10 shows the average multiclass classification performance based on the NSL-KDD dataset with AE-WGAN architecture.

The AE-WGAN model, focused on producing high-quality attacks, was evaluated over different attack categories to reveal its detection efficiency.

The performance of AIDS over all classifiers was decreased in multiclass classification, where the AIDS performance became more challenging to classify four attack classes (DoS, Probe, R2L, and U2R) compared to two classes (Normal, Attack) in binary classification. The experiments proved that the DL classifier performed ideally compared with ML classifiers. Regarding F1 score, the RNN and GRU were the best compared to DT and RF classifiers. Also, the LSTM and CNN were better than ML classifiers.

The results were validated by focusing on rare attacks of AIDS proposed based on AE-WGAN architecture. Therefore, Probe, R2L, and U2R were investigated over the mentioned classifiers. The results showed that the CNN classifier based on AE-WGAN outperformed the rest models, especially for rare attacks except the U2R category. Although the model performed perfectly on different attack categories, the insufficient samples in the U2R category, even with adversarial attacks, did not help since the learning behaviour was deficient. Fig. 11 shows the AIDS based on CNN over the attack categories.

To verify the superiority of the proposed architecture, the AE-WGAN was compared with WGAN architecture. Table 6 shows the improvements achieved in terms of the F1 score where AIDS was implemented by the CNN classifier and the adversarial attacks obtained by AE-WGAN.

According to Table 6 the F1 scores in AIDS enhanced significantly for R2L with a 37.0 %, which consists of rare attacks compared with the DoS and Probe categories. Also, DoS and Probe were improved by 8.0 %, 2.0 %, respectively. However, the AIDS performance for the U2R category was limited because the AE-WGAN did not learn from limited samples, which led to limited attack detection.

5.4.2. Model performance on CICIDS-2017

AIDS AE-WGAN architecture based was evaluated on the CICIDS-2017 dataset. In binary classification, the revealed results outperformed DL compared to ML classifiers in evaluation metrics. For DL classifiers, the F1 scores reached 98.0 % for CNN, GRU, LSTM, and RNN, while the ML did not exceed 84.0 % for DT and 88.0 % for RF classier. The most outstanding precision score was 99.0 % using the GRU, while the highest in DT and RF was 79.0 %. Fig. 12 shows the AIDS performance based on AE-WGAN architecture with the CICIDS-2017 dataset.

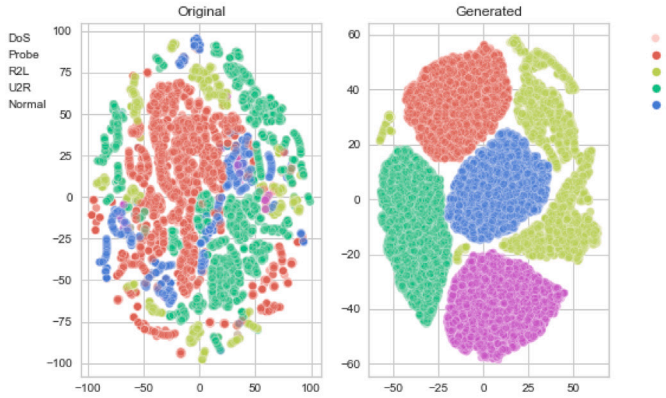
Referring to Fig. 12, the results proved that AIDS based on AE-WGAN architecture delivered high performance in attack detection even with many attack types, such as the CICIDS-2017 dataset. Also, the performance in NSL-KDD for binary classification was high, but the number of samples for rare attacks in the training set affected AIDS in attack detection.

For more application, AIDS based on AE-WGAN was evaluated on multiclass classification and the CICIDS-2017 dataset. The performance results validated that the DL was more appropriate for delivering high performance than the ML classifiers. Regarding the F1 score, the GRU and LSTM classifiers reached 69.0 % compared to RNN and CNN, which reached 66.0 %, and 65.0 %, respectively. However, the F1 score value was 52.0 % in DT and 63.0 % in RF classifier. Fig. 13 shows AIDS performance based on AE-WGAN and the CICIDS-2017 dataset in multiclass classification.

The performance of AIDS in CICIDS-2017 varied in multiclass classification, where the GRU classifier performed perfectly in attack detection. In addition, the AE-WGAN enhanced the model by detecting ten attack types. The most significant improvement of detection in terms of F1 score was 40.0 %, 38.0 %, 26.0 %, 17.0 %, and 11.0 % for DoS GoldenEye, SQL Injection, DoS Slowhttptest, SSH Patator, and FTP Patator attacks, respectively. Also, four attack types were detected by the GRU classifier, where the F1 score was 8.0 %, 7.0 %, 6.0 %, 2.0 %, and 2.0 % for Heartbleed, Brute Force, Bot, DDoS, and Port Scan attacks, respectively. In contrast, DoS Hulk, DoS slowloris, Infiltration, and XSS were not detected, as the number of training samples was a few compared to other attack types. Table 7 shows the amount of improvement for AIDS based on the AE-WGAN with the CICIDS-2017 dataset.

The percentage improvement was calculated by comparing the F1 scores of the AE-WGAN model with those of the baseline WGAN model

(a) NSL-KDD dataset



(b) CICIDS-2017 dataset

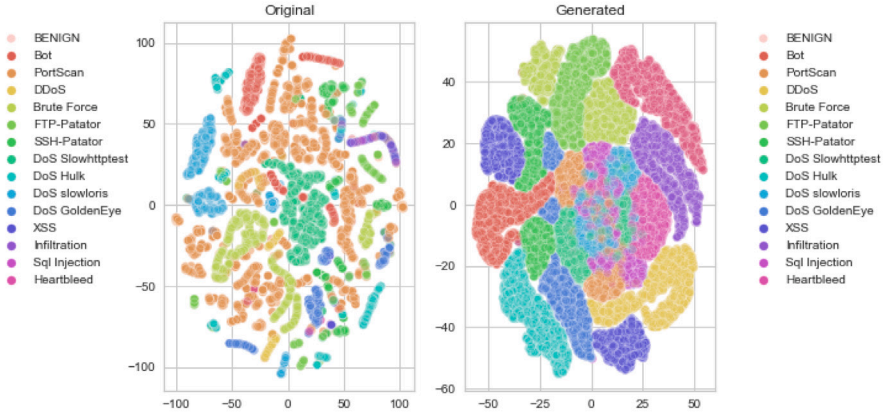
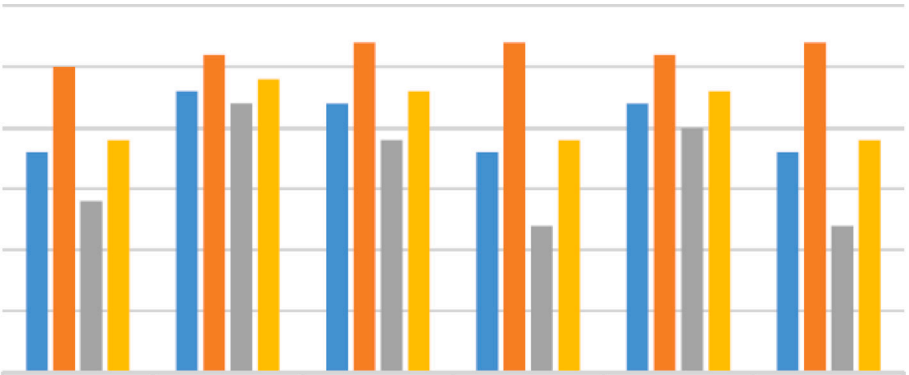
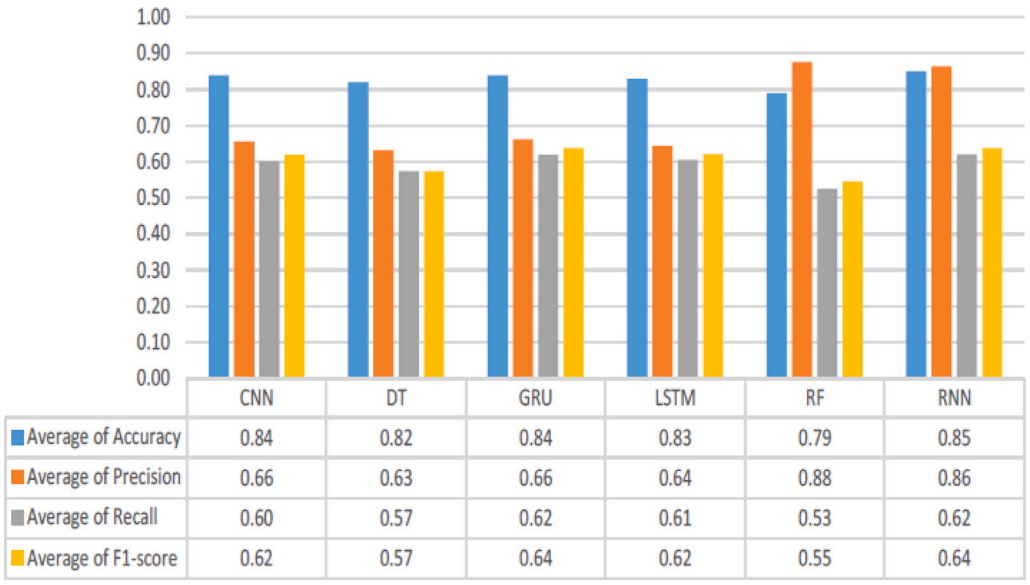


Fig. 8. Data distributions of datasets before and after applying AE-WGAN architecture.



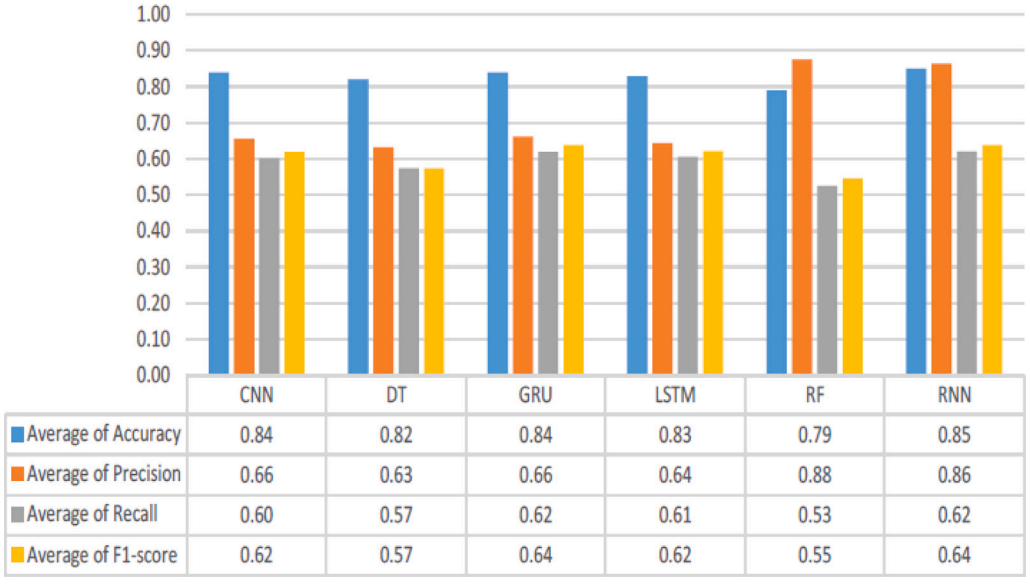
	CNN	DT	RF	GRU	LSTM	RNN
Accuracy	0.88	0.93	0.92	0.88	0.92	0.88
Precision	0.95	0.96	0.97	0.97	0.96	0.97
Recall	0.84	0.92	0.89	0.82	0.9	0.82
F1-score	0.89	0.94	0.93	0.89	0.93	0.89

Fig. 9. The performance of AIDS in binary classification based on NSL-KDD and AE-WGAN architecture.



	CNN	DT	GRU	LSTM	RF	RNN
Average of Accuracy	0.84	0.82	0.84	0.83	0.79	0.85
Average of Precision	0.66	0.63	0.66	0.64	0.88	0.86
Average of Recall	0.60	0.57	0.62	0.61	0.53	0.62
Average of F1-score	0.62	0.57	0.64	0.62	0.55	0.64

Fig. 10. The performance of AIDS in multiclass classification based on NSL-KDD and AE-WGAN architecture.



	CNN	DT	GRU	LSTM	RF	RNN
Average of Accuracy	0.84	0.82	0.84	0.83	0.79	0.85
Average of Precision	0.66	0.63	0.66	0.64	0.88	0.86
Average of Recall	0.60	0.57	0.62	0.61	0.53	0.62
Average of F1-score	0.62	0.57	0.64	0.62	0.55	0.64

Fig. 11. The performance of AIDS in multiclass classification based on AE-WGAN.

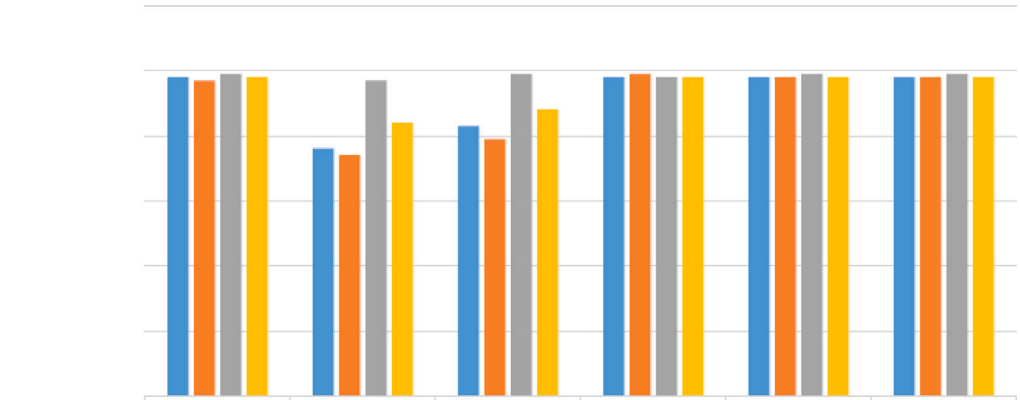
Table 6
AIDS performance on NSL-KDD using CNN classifier.

Category	Accuracy		Precision		Recall		F1		Improvement
	WGAN	AE-WGAN	WGAN	AE-WGAN	WGAN	AE-WGAN	WGAN	AE-WGAN	%
Normal	0.77	0.84	0.80	0.81	0.95	0.97	0.86	0.89	2.0
DoS	0.77	0.84	0.79	0.92	0.85	0.87	0.81	0.89	**8.0**
Probe	0.77	0.84	0.80	0.77	0.56	0.61	0.66	0.68	**2.0**
R2L	0.77	0.84	0.80	0.78	0.17	0.56	0.28	0.65	**37.0**
U2R	0.77	0.84	0.06	0.0	0.22	0.0	0.09	0.0	−9.0
All	0.89	0.88	0.93	0.95	0.88	0.84	0.90	0.89	−1.0

for each attack type. The formula used is: $Improvement = (F1_AE - WGAN - F1_WGAN) * 100$.

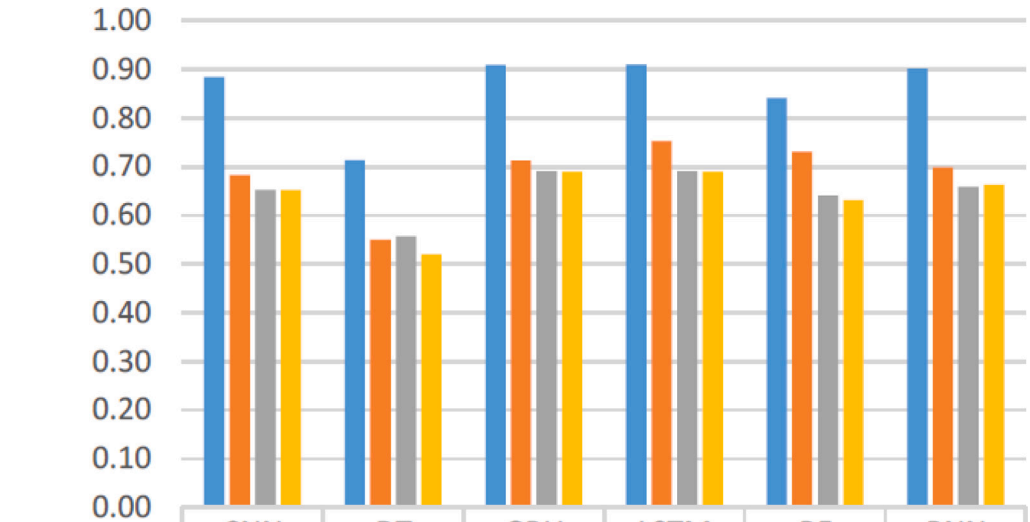
Fig. 14 presents the Receiver Operating Characteristic (ROC) curves for binary classification using the AE-WGAN model with GRU classifier on the NSL-KDD and CICIDS-2017 datasets. These curves illustrate the model's ability to distinguish between normal and attack instances at various classification thresholds.

The ROC curves in Fig. 14 demonstrate the exceptional performance of the AE-WGAN model with GRU classifier for binary classification on both NSL-KDD and CICIDS-2017 datasets. For NSL-KDD, the model achieves AUC values of 0.99 and 0.89 for training and test sets respectively, while for CICIDS-2017, it attains even higher AUC values of 0.99 and 0.97. These near-perfect AUC scores indicate the model's robust ability to distinguish between normal and attack instances with



	CNN	DT	RF	GRU	LSTM	RNN
Accuracy	0.98	0.76	0.83	0.98	0.98	0.98
Precision	0.97	0.74	0.79	0.99	0.98	0.98
Recall	0.99	0.97	0.99	0.98	0.99	0.99
F1-score	0.98	0.84	0.88	0.98	0.98	0.98

Fig. 12. The performance of AIDS in binary classification based on AE-WGAN.



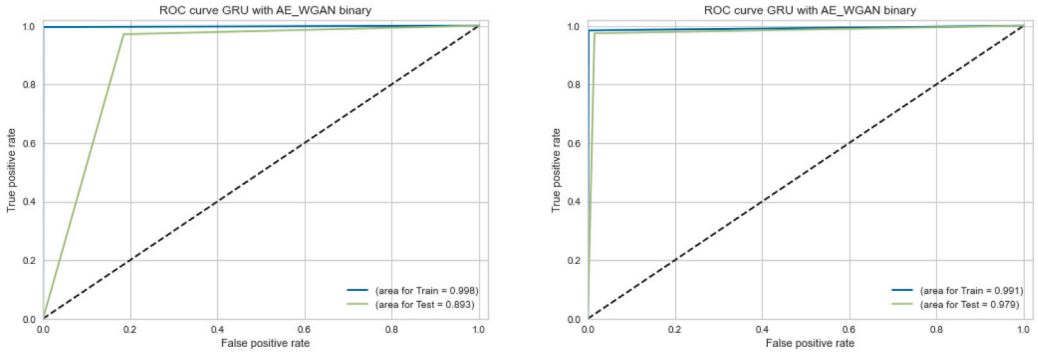
	CNN	DT	GRU	LSTM	RF	RNN
Accuracy	0.88	0.71	0.91	0.91	0.84	0.90
Precision	0.68	0.55	0.71	0.75	0.73	0.70
Recall	0.65	0.56	0.69	0.69	0.64	0.66
F1	0.65	0.52	0.69	0.69	0.63	0.66

Fig. 13. The performance of AIDS in multiclass classification based on AE-WGAN.

minimal false positives. The consistent high performance across both datasets and between training and test sets underscores the model's effectiveness and generalizability in intrusion detection tasks. This suggests that the AE-WGAN approach can maintain high true positive rates while keeping false positives low, a crucial characteristic for practical deployment in diverse network security environments.

5.5. Generalization comparison with existing models

To verify the AE-WGAN model's distinction, the study compared the proposed model with the state-of-the-art models. For a fair comparison, all models built based on WGAN and datasets (NSL-KDD, CICIDS-2017) were selected. Also, standard evaluation metrics were included



(a) NSL-KDD dataset (b) CICIDS-2017 dataset

Fig. 14. ROC curves for binary classification using the AE-WGAN model with GRU classifier on (a) NSL-KDD and (b) CICIDS-2017 datasets.

Table 7
AIDS performance on CICIDS-2017 using GRU classifier.

Attack name	F1(AE-WGAN)	F1(WGAN)	Improvement (F1) %
BENIGN	0.97	0.95	**2.00**
Bot	0.82	0.76	**6.00**
Brute Force	0.97	0.90	**7.00**
DDoS	0.99	0.97	**2.00**
DoS GoldenEye	0.40	0.00	**40.00**
DoS Hulk	1.00	1.00	0.00
DoS Slowhttptest	0.93	0.67	**26.00**
DoS slowloris	0.00	0.00	0.00
FTP Patator	0.97	0.86	**11.00**
Heartbleed	0.09	0.00	**8.60**
Infiltration	0.99	0.99	0.00
PortScan	0.66	0.64	**2.00**
Sql Injection	0.38	0.00	**38.00**
SSH Patator	0.95	0.78	**17.00**
XSS	0.99	0.99	0.00

for generalization assessments. Table 8 shows the AE-WGNA model performance compared with state-of-the-art models.

Table 8 shows that the AIDS based on AE-WGAN outperformed the state-of-the-art models in terms of Accuracy, Precision, Recall, and F1 metrics on both datasets. AE-WGAN performance compared with the existing models showed that the generated attacks produced with high quality, which was used to tackle imbalanced datasets. In particular, the AE-WGAN IDS base showed that the F1 and Recall metrics were enhanced significantly, as these metrics were used for an imbalanced dataset issue. Therefore, the robustness of AID based on the AE-WGAN architecture was proven against minority and majority attack classes.

The AE-WGAN model outperforms baseline models due to several key factors:

- Enhanced feature extraction: The denoising autoencoder component allows for more robust and informative feature representation.
- Improved synthetic attack quality: The combination of AE and WGAN results in more realistic and diverse synthetic attacks, enhancing the model's ability to detect rare attack types.
- Better handling of imbalanced data: The model's ability to generate high-quality synthetic samples for minority classes addresses the class imbalance issue more effectively than previous approaches.
- Stability in training: The use of Wasserstein distance in the GAN component leads to more stable training and better convergence compared to traditional GANs.
- These factors collectively contribute to the model's superior performance across various metrics and datasets.

In terms of computational requirements, the AE-WGAN model took an average of 2.5 h to train on the NSL-KDD dataset and 3.8 h on the CICIDS-2017 dataset using a single NVIDIA RTX 3080 GPU. Inference time averaged 0.05 s per sample, making it suitable for real-time intrusion detection in practical network environments. This computational efficiency, combined with its high performance, makes the approach a viable solution for large-scale deployment.

5.6. Generalization comparison with existing models

To rigorously verify the distinction of the AE-WGAN model, the study conducted a comprehensive comparison with state-of-the-art models. For a fair comparison, models built based on WGAN were selected and evaluated on the NSL-KDD and CICIDS-2017 datasets. Multiple runs of each model were performed to obtain mean performance metrics and standard deviations, allowing for statistical significance testing. Multiple runs of each model were performed to obtain mean performance metrics and standard deviations, allowing for statistical significance testing. Table 9 shows the detailed performance comparison of the AE-WGAN model with state-of-the-art models.

To assess the statistical significance of the model's improvements, paired t-tests were conducted comparing the AE-WGAN model with each baseline method. A significance level of $\alpha = 0.05$ was used. The results of these statistical tests are as follows:

- AE-WGAN vs CWGAN-CSSAE on NSL-KDD: t-statistic = 12.37, p-value = 3.2e−5 (significant)
- AE-WGAN vs GMM-WGAN-IDS on NSL-KDD: t-statistic = 9.81, p-value = 1.8e−4 (significant)
- AE-WGAN vs AE on NSL-KDD: t-statistic = 15.23, p-value = 7.9e−6 (significant)
- AE-WGAN vs HMCD-Model on CICIDS-2017: t-statistic = 11.05, p-value = 5.7e−5 (significant)

These results demonstrate that the AE-WGAN model achieves statistically significant improvements overall baseline methods across both datasets. The most notable improvements are:

- On NSL-KDD: 6.82% increase in F1-score compared to CWGAN-CSSAE, 7.12% increase compared to GMM-WGAN-IDS, and 12.79% increase compared to AE.
- On CICIDS-2017: 14.34% increase in F1-score compared to HMCD-Model.

To further illustrate the effectiveness of the approach, especially in handling imbalanced datasets, the study analysed the performance improvements for specific attack categories. Table 10 shows the comparison of F1-scores for different attack types on the NSL-KDD dataset.

Table 8
Comparison results of AE-WGAN with state-of-the-art models.

Models	Dataset	Accuracy	Precision	Recall	F1
CWGAN-CSSAE [7]	NSL-KDD	80.78	95.98	79.86	87.18
GMM-WGAN-IDS [15]	NSL-KDD	86.59	88.55	86.59	86.88
AE [8]	NSL-KDD	–	87.85	82.04	81.21
AE-WGAN	NSL-KDD	**93.00**	**96.00**	**92.00**	**94.00**
HMCD-Model [12]	CICIDS-2017	–	96.62	73.77	83.66
AE-WGAN	CICIDS-2017	**98.00**	**98.00**	**99.00**	**98.00**

Table 9
Detailed comparison of AE-WGAN with state-of-the-art models (mean ± standard deviation).

Model	Dataset	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
AE-WGAN (Ours)	NSL-KDD	**93.00 ± 0.42**	**96.00 ± 0.38**	**92.00 ± 0.51**	**94.00 ± 0.45**
CWGAN-CSSAE [7]	NSL-KDD	80.78 ± 0.65	95.98 ± 0.47	79.86 ± 0.72	87.18 ± 0.58
GMM-WGAN-IDS [15]	NSL-KDD	86.59 ± 0.53	88.55 ± 0.49	86.59 ± 0.61	86.88 ± 0.55
AE [8]	NSL-KDD	–	87.85 ± 0.57	82.04 ± 0.68	81.21 ± 0.62
AE-WGAN (Ours)	CICIDS-2017	**98.00 ± 0.31**	**98.00 ± 0.29**	**99.00 ± 0.27**	**98.00 ± 0.30**
HMCD-Model [12]	CICIDS-2017	–	96.62 ± 0.41	73.77 ± 0.79	83.66 ± 0.58

Table 10
Comparison of F1-scores for specific attack categories on NSL-KDD.

Attack category	AE-WGAN F1-score (%)	Best Baseline F1-score (%)	Improvement (%)
DoS	**89.0**	81.0 (CWGAN-CSSAE)	+**8.0**
Probe	**68.0**	66.0 (GMM-WGAN-IDS)	+**2.0**
R2L	**65.0**	28.0 (CWGAN-CSSAE)	+**37.0**
U2R	0.0	**9.0** (CWGAN-CSSAE)	−9.0

The substantial improvement in R2L detection (37% increase in F1-score) the model's strength in handling rare attack types. However, the challenge with U2R attacks remains an area for future work.

These detailed comparisons and statistical analyses provide strong evidence for the superiority of the AE-WGAN approach over existing state-of-the-art methods, particularly in addressing the challenges of imbalanced datasets and rare attack detection in network intrusion detection systems. The significant improvements across various metrics and attack types validate the effectiveness of the proposed architecture in enhancing the overall performance of anomaly-based intrusion detection systems.

The experimental results demonstrate the effectiveness of the proposed AE-WGAN architecture in improving intrusion detection performance. Key findings include:

- Significant improvement in detecting rare attack types, particularly R2L attacks.
- Consistent performance enhancement across both binary and multi-class classification tasks.
- Superior generalization capability against unseen attacks.
- Effectiveness in handling imbalanced datasets through high-quality synthetic attack generation.
- These results suggest that the AE-WGAN approach offers a robust solution to the challenges of high-dimensionality, large-scale data, and class imbalance in network intrusion detection systems.

While this study focuses on the NSL-KDD and CICIDS-2017 datasets, these benchmarks represent a diverse range of attack types and network traffic patterns. NSL-KDD contains 22 attack types across 4 major categories, while CICIDS-2017 includes 15 modern attack types simulated in a diverse network environment. The strong performance across both datasets demonstrates the AE-WGAN model's ability to generalize to different attack scenarios and network topologies. Future work could further validate the model's generalizability by evaluating it on additional datasets with unique characteristics, such as the UNSW-NB15 dataset for modern attack types or the IoT-23 dataset for IoT-specific intrusion detection scenarios.

6. Conclusion

This paper presents a novel approach to anomaly-based intrusion detection using a combination of denoising autoencoders and Wasserstein GANs. The proposed AE-WGAN architecture effectively addresses key challenges in network intrusion detection, including high dimensionality, large-scale data, and class imbalance. By leveraging the ANOVA method for feature selection and the AE-WGAN for generating high-quality synthetic attacks, the model achieves superior performance in detecting both common and rare attack types. Extensive experiments on the NSL-KDD and CICIDS-2017 datasets demonstrate the model's effectiveness across various classification scenarios and its ability to outperform state-of-the-art approaches. The improved detection rates for minority attack classes, such as U2R and R2L, highlight the model's potential for enhancing security in real-world network environments.

Future research avenues for enhancing the proposed AE-WGAN model encompass several key areas. These include the exploration of transfer learning techniques to facilitate adaptation to novel attack types, the integration of attention mechanisms to bolster model interpretability, and the development of an online learning variant to address evolving network threats. Additionally, extending the model's capabilities to process multi-modal input data, incorporating both flow statistics and packet payload information, presents an intriguing direction. Finally, extensive real-world deployments across diverse network environments are crucial to rigorously validate the model's efficacy and generalizability in practical setting

CRediT authorship contribution statement

Mohammad Arafah: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Iain Phillips:** Supervision. **Asma Adnane:** Supervision. **Wael Hadi:** Validation, Conceptualization. **Mohammad Alauthman:** Validation. **Abedal-Kareem Al-Banna:** Validation.

Declaration of competing interest

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

Data availability

Data will be made available on request.

References

[1] Z. Wang, Y. Liu, D. He, S. Chan, Intrusion detection methods based on integrated deep learning model, Comput. Secur. 103 (2021) 102177.

[2] M. Ammar, G. Russello, B. Crispo, Internet of Things: A survey on the security of IoT frameworks, J. Inf. Secur. Appl. 38 (2018) 8–27.

[3] X. Xu, J. Li, Y. Yang, F. Shen, Toward effective intrusion detection using log-cosh conditional variational autoencoder, IEEE Internet Things J. 8 (8) (2020) 6187–6196.

[4] C. Kreibich, J. Crowcroft, Honeycomb: creating intrusion detection signatures using honeypots, ACM SIGCOMM Comput. Commun. Rev. 34 (1) (2004) 51–56.

[5] A.L. Buczak, E. Guven, A survey of data mining and machine learning methods for cyber security intrusion detection, IEEE Commun. Surv. Tutor. 18 (2) (2016) 1153–1176.

[6] J. Yang, T. Li, G. Liang, Y. Wang, T. Gao, F. Zhu, Spam transaction attack detection model based on GRU and WGAN-div, Comput. Commun. 161 (2020) 172–182.

[7] G. Zhang, X. Wang, R. Li, Y. Song, J. He, J. Lai, Network intrusion detection based on conditional wasserstein generative adversarial network and cost-sensitive stacked autoencoder, IEEE Access 8 (2020) 190431–190447.

[8] C. Ieracitano, A. Adeel, F.C. Morabito, A. Hussain, A novel statistical analysis and autoencoder driven intelligent intrusion detection approach, Neurocomputing 387 (2020) 51–62.

[9] J. Lee, K. Park, AE-CGAN model based high performance network intrusion detection system, Appl. Sci. 9 (20) (2019) 4221.

[10] M.O. Kaplan, S.E. Alptekin, An improved BiGAN based approach for anomaly detection, Procedia Comput. Sci. 176 (2020) 185–194.

[11] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, A. Courville, in: I. Guyon, et al. (Eds.), Advances in Neural Information Processing Systems 30 (NIPS 2017), Vol. 5767, Curran Associates, Inc, Red Hook, NY, 2017.

[12] X. Yun, J. Xie, S. Li, Y. Zhang, P. Sun, Detecting unknown HTTP-based malicious communication behavior via generated adversarial flows and hierarchical traffic features, Comput. Secur. 121 (2022) 102834.

[13] M. Tavallaee, E. Bagheri, W. Lu, A.A. Ghorbani, A detailed analysis of the KDD cup 99 data set, in: 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, Ieee, 2009, pp. 1–6.

[14] Y. Liu, S. Garg, J. Nie, Y. Zhang, Z. Xiong, J. Kang, M.S. Hossain, Deep anomaly detection for time-series data in industrial IoT: A communication-efficient on-device federated learning approach, IEEE Internet Things J. 8 (8) (2021) 6348–6358.

[15] J. Cui, L. Zong, J. Xie, M. Tang, A novel multi-module integrated intrusion detection system for high-dimensional imbalanced data, Appl. Intell. (2022) 1–17.

[16] A.I. Gide, A.A. Mu'azu, A real-time intrusion detection system for dos/ddos attack classification in IoT networks using KNN-neural network hybrid technique, Babylonian J. Int. Things 2024 (2024) 60–69.

[17] D. Saxena, J. Cao, Generative adversarial networks (GANs) challenges, solutions, and future directions, ACM Comput. Surv. 54 (3) (2021) 1–42.

[18] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, Adv. Neural Inf. Process. Syst. 27 (2014).

[19] M. Arjovsky, S. Chintala, L. Bottou, Wasserstein generative adversarial networks, in: D. Precup, Y.W. Teh (Eds.), Proceedings of the 34th International Conference on Machine Learning, in: Proceedings of Machine Learning Research, Vol. 70, PMLR, 2017, pp. 214–223, URL https://proceedings.mlr.press/v70/arjovsky17a.html.

[20] G.E. Hinton, R.R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science 313 (5786) (2006) 504–507.

[21] M. Arafah, AE-WGAN: Anomaly-based network intrusion detection using denoising autoencoder and wasserstein GAN, 2023, https://github.com/marafah/AE-WGAN.git.

[22] I. Sharafaldin, A.H. Lashkari, A.A. Ghorbani, Intrusion detection evaluation dataset (CIC-ids2017), in: Proceedings of the of Canadian Institute for Cybersecurity, 2018.

[23] J.R. Quinlan, Induction of decision trees, Mach. Learn. 1 (1) (1986) 81–106, http://dx.doi.org/10.1007/BF00116251, URL https://link.springer.com/article/10.1007/BF00116251.

[24] L. Breiman, J.H. Friedman, R.A. Olshen, C.J. Stone, Classification and Regression Trees, Wadsworth and Brooks/Cole Advanced Books & Software, Monterey, CA, 1984.