Journal of Information Systems Engineering and Management
2025, 10(16s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Wasserstein Conditional Generative Adversarial Networks for Class Balancing in Intrusion Detection Datasets

Lina Aziz Swadi*[1], Haider M. Al-Mashhadi[2]
[1]*Department of Computer Science, College of Computer Science and Information Technology University of Basrah, Basrah, Iraq*
[1]*Department of Cybersecurity, College of Computer Science and Information Technology University of Basrah, Basrah, Iraq*
[1]liaz68@yahoo.com, [2]haider.abdulnabi@uobasrah.edu.iq

ARTICLE INFO

Received: 02 Dec 2024
Revised: 22 Jan 2025
Accepted: 06 Feb 2025

ABSTRACT

Conventional network intrusion detection systems have numerous obstacles in handling data. These issues may significantly impact its efficacy and efficiency. In real-world scenarios, attacks are rare compared to the high volume of normal network activities, creating a significant imbalance in the dataset. This imbalance causes the model to concentrate more on normal traffic data, thus decreasing its sensitivity in identifying attacks and impacting the overall efficacy of the intrusion detection system. Training data plays a critical role when training an intrusion detection system. Generating enough training data is a challenging task. One approach to handle this challenge is to utilize Generative Adversarial Networks, a machine learning technique that generates synthetic data by placing a generator and a discriminator—two neural networks contradict each other. The generator creates realistic data examples, while the discriminator assesses them, improving the data's validity through iterative training. This paper suggests employing Wasserstein Conditional Generative Adversarial Networks (WCGANs) to tackle the imbalanced class issue and improve the effectiveness of intrusion detection systems. Providing realistic adversarial examples, the model enhances deep neural network training, hence complementing deep learning techniques. This research focuses on handling the difficulty of class imbalance in network intrusion detection models using WCGAN. By generating synthetic data for both normal and attack categories, the system improves the detection of underrepresented labels. WCGAN can also be leveraged to generate realistic network traffic samples, thus enhancing the robustness of the classifiers in both binary and multi-class scenarios.

Keywords: ANN, GANs, Imbalance dataset, synthetic dataset, IDS.

1. INTRODUCTION

The information technology community and the underlying digital platform have collaborated in the overwhelming majority of business sectors and societal aspects. This digitization has, in parallel, raised significant challenges for cybersecurity[1]. The estimation is that threat events are growing at the same pace as developments in connectivity, mobility, and diversity. The intrusion incidents are noise within the system information, significantly impacting the core of enterprise services and, on a larger scale, nation-state support services [2]. Detecting intrusions and malicious activity is a key challenge in cybersecurity. Lately, researchers have concentrated on utilizing artificial intelligence techniques for Network Intrusion Detection System (NIDS). The AI-based intrusion detection system has shown outstanding performance. Initially, researchers focused on incorporating traditional machine learning approaches like SVM and decision trees into the currently available Intrusion Detection System (IDS). This has now widened to encompass deep learning methods like CNN, LSTM, and autoencoders. However, these methods have limitations in their real-world application [3].

The trouble with machine learning models is determined as the non-stationary threat datasets with only a few patterns. In addition to optimizing models that reduce false alarms, a structured framework must be developed to improve intrusion detection systems to handle large and imbalanced datasets effectively [4]. A novel machine learning framework was introduced that incorporates Random Oversampling (RO), Stacking Feature Embedding (SFE), and

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Principal Component Analysis (PCA) to handle the difficulties posed by big and imbalanced data, enhancing detection accuracy and robustness [5].

Class imbalance presents a considerable obstruction in NIDS, as normal traffic data significantly exceeds attack data or vice versa, resulting in diminished detection accuracy. To handle this issue, Generative Adversarial Networks (GANs) were employed to create artificial data samples, thus balancing the dataset across diverse classes.

The generated samples are utilized to train machine learning algorithms, enhancing their effectiveness in identifying both prevalent and infrequent attack types, which is considered a novel methodology for detecting intrusions via Generative Adversarial Networks [6].

Big Data exploration concepts via data generation using Generative Deep Models are useful in assisting the acquisition of a generalizable decision boundary. A model preferably allows the intrusion and benign classes to be easily separated and deal with imbalanced classes [7]. SAGANs of Zhang et al. (2022) suggested using a pair of generator and discriminator networks encoded by attention mechanisms to encourage predictability, which can generate high-quality samples and learn meaningful representations on training data and testing data [8].

GANs have been introduced as powerful gadgets for producing synthetic data in various fields. Conditional GANs extend this capability by generating data conditioned on labels, which can be specifically useful for classification tasks [9]. The current paper utilizes a hybrid generative model combining Wasserstein GAN and Conditional GAN to generate a novel dataset that has similar data patterns and distribution as the original dataset with reasonable similarity [10].

Two Wasserstein Conditional Generative Adversarial Networks (WCGANs) models were utilized to create balanced artificial data for binary and multiclass classification. The initial WCGAN model is utilized for binary classification, distinguishing network traffic as normal or an assault. The other WCGAN model is used for multiclass classification to balance the dataset among various attack categories by generating synthetic samples in underrepresented categories. The system is experimented with utilizing the UNSW-NB15 and KDD CUP99 datasets for evaluation. The model aims to process the class imbalance challenge often faced in intrusion detection datasets.

2. BACKGROUND

2.1 Artificial Neural Network (ANN)

An artificial neural network (ANN) is a shape of artificial intelligence that aims to emulate the functioning of a human being's brain. It is considered effective in predicting events when studying a massive amount of data. Networks are implemented based on calculations and parameters essential for predicting the outcome. As shown in Figure 1. [11], it comprises processing nodes (units) arranged into input, hidden, and output layers. The units in every layer are interconnected with units in the next layers. Every connection possesses a weight value. The inputs are multiplied by their corresponding weights and aggregated at every node. The total subsequently undergoes a transformation dictated by the activation function, generally a sigmoid function, rectified linear unit (ReLU), or hyperbolic tangent. In feedforward networks, information progresses from the input to the output layer, perhaps traversing hidden nodes. Backpropagation occurs by comparing the output with the target output and computing the discrepancy known as the error function. This error is transmitted backward through each network layer to modify the weights to reduce the error. This operation is repeated several times till the network's performance is enhanced [12].

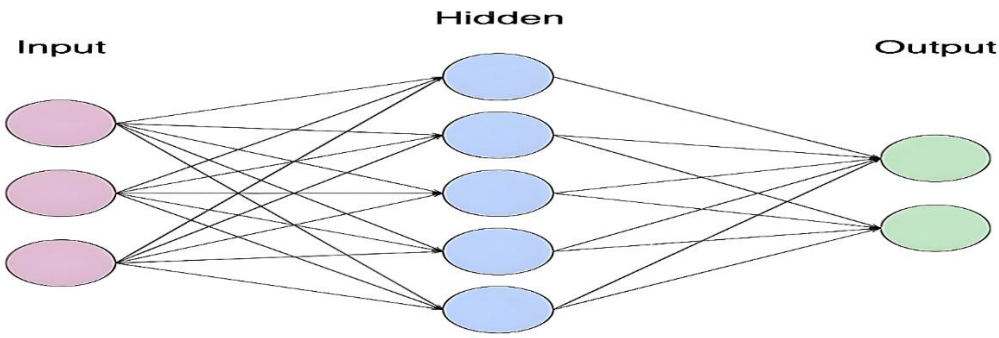

Fig. 1. Architecture of Artificial Neural Network (ANN)

2.2 Generative Adversarial Networks (GANs)

Generative Adversarial Networks (GANs) were first introduced by Goodfellow et al. (2014) and consist of two distinct models: Generator and Discriminator. GANs attempt to simulate the distribution of real data by adversarially training a generative network, which infers a probabilistic mapping from the real data space to the input space, and a discriminative network, which predicts the probability that a given data example originated from the generative network rather than the real data. The generative model, trained to optimize the likelihood of error in the discriminative model, attempts to create data identical to the original data. The discriminative model, trained to minimize the likelihood of attributing significant importance to anything other than examples coming from the real data, attempts to distinguish between fake data generated using the generative model as a training signal and real data [13].

The generator uses a random noise vector as input. The discriminator receives data input from authentic or generated data samples using binary cross-entropy loss to determine if the input data is fake or real. As shown in Figure 2 [14], the activation function [15] is given by:

$$G_{min}D_{max}V(G,D) = E_{x\sim pdata(x)}\left[logD(x)\right] + E_{z\sim pz(z)}\left[\log\left(1 - D\left(G(z)\right)\right)\right] \quad (1)$$

where x: is real data, z: is a latent space vector, G(z): is produced data, whereas D(x): represents the assessment of authentic data by the discriminator, and D(G(z)): represents the assessment of generated data by the discriminator.

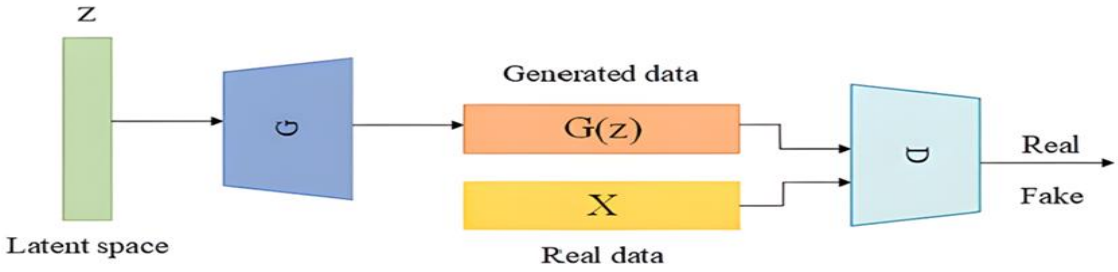

Fig. 2. Architecture of Generative Adversarial Networks (GANs)

2.3 Conditional (cGANs) Overview

It is another kind of GANs where the generation operation is conditioned on further information, such as attributes, class labels, or other contextual data. This conditioning permits cGANs to produce samples that meet specific characteristics, making them effective in generating realistic samples for targeted scenarios [16].

However, adversarial training of CGANs can be challenging. Due to the nature of the min-max enhancement, solutions may exhibit oscillations, and instabilities, or fail to converge. This sensitivity is highly affected by the structure of the models and the chosen hyperparameters. Additionally, traditional evaluation metrics like maximum likelihood estimates are less meaningful in assessing CGAN performance due to independence assumptions, which can result in underestimating the data probability.

Therefore, alternative evaluation approaches are often required to accurately assess the quality and stability of CGAN outputs [17]. As shown in Figure 3 [18]. The activation function [19] is given by:

$$G_{min}\ D_{max}\ V(G,D) = E_{x\sim pdata(x)}\left[logD(x|y)\right] + E_{z\sim pz(z)}\left[lo\ g\left(1 - D\left(G(z|y)\right)\right)\right] \quad (2)$$

where y: represents the class label, D(x|y): the probability estimate generated via the discriminator that x is a real example, contingent upon the given label y, G(z|y): is the generator's output, conditioned on y, which tries to create examples resembling x based on the label y, D(G(z|y)): depict how the discriminator assess data created by the generator conditioned on label y.

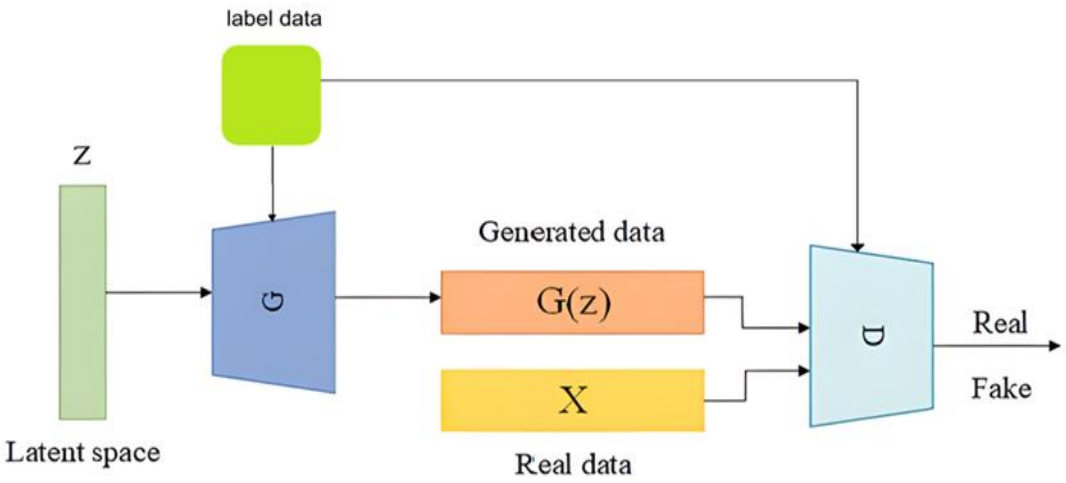

Fig. 3. Architecture of Conditional Generative Adversarial Networks (cGAN)

2.4 WASSERSTEIN GANs OVERVIEW

It is presented by Arjovsky et al. (2017), improved upon standard GANs by utilizing the wasserstein distance as a loss function, facilitating improved convergence properties [20]. It is a variant of GANs that utilizes the Wasserstein distance to enhance training stability and performance. It also involves measuring the distance among probability distributions of the real data and the generated data, providing a more significant gradient for the generator and discriminator training. This technique enhances the model's capability to generate realistic samples [21]. WGAN enables a robust discriminator that provides valuable gradients to the generator even in cases when the quality of the reproduced instances is still limited, hence enhancing the stability of training [22].

3. RELATED WORK

Authors in [23] proposed a distributed-based GAN network to generate 10,000 synthetic data for each class label (normal and attacks) equally distributed and use a boxplot to investigate the quality of the artificial data, guaranteeing a more precise depiction of real-world data distribution. The reproduced data is utilized to train machine learning techniques although the results are competitive with those obtained from real datasets. There is a possibility that networks trained exclusively on artificial data may not generalize well to all real-world scenarios. This may result in performance discrepancies when deployed in live environments.

The study in [24] created and implemented a cGANs model to balance and amplify input for both normal and malignant labels. Application of the ocGAN model in frameworks for anomaly identification in imbalanced datasets and data balancing. Frameworks for anomaly detection and data augmentation utilizing the Bayesian Convolutional Neural Network (bcGAN) paradigm. This paper employed a multiclass classification system for data augmentation and identification of anomalies, utilizing the bcGAN model. The study suggested a deep learning method for the detection of anomalies in IoT ecosystems utilizing synthetic data produced by cGANs. The enhancement of performance in the detection of anomalies throughout diverse IoT networks. However, an inherent constraint of ocGAN and bcGAN models is their tendency to achieve inferior detection rates when the size of the training data sample is below 1000. Higher detection rates were seen when the sample size of the training data exceeded 1000.

In [25] the work suggested two different types of generative models, Bidirectional Generative Adversarial Networks (BiGAN) and Adversarial Autoencoders (AAE), to produce artificial data for training the IDS. The effectiveness of the models was assessed using a method called stratified 10-fold cross-validation. The generative models, particularly the BiGAN and AAE, performed significantly better than traditional machine learning approaches, such as Random Forests in identifying cyberattacks. These models achieved high F1-scores (up to 0.99) when classifying various types of attacks, indicating their potential to optimize the accuracy and robustness of IDS in IoT environments. However, their performance on datasets other than IoT-23 or in real-world situations has not been tested. Generative models are intricate and may be susceptible to overfitting if they are not appropriately regulated or if they are trained using limited or unrepresentative data. The effectiveness of models depends significantly on the quality and diversity of training data. Biases in the dataset can lead to poor adaptation to real-world conditions.

In [26] the research employed a Conditional Tabular Generative Adversarial Network (CTGAN) within an Intrusion Detection System (IDS) to enhance the detection of DoS and DDoS attacks. This method involves the generation of emulated traffic that closely resembles normal traffic patterns, aiding in the differentiation between legitimate and malicious activities. Research findings indicated that the CTGAN-based IDS can effectively identify DDoS and DoS attacks within IoT networks. Additionally, the synthetic data produced by CTGAN contributes to the improved training of machine learning models, bolstering their capacity to detect attacks. However, it is significant to note that the suggested IDS may encounter challenges in identifying new or modified attack vectors as malicious actors constantly adapt their tactics, posing difficulty in maintaining efficacy. Furthermore, the results of this paper may not be universally applicable to other network types or diverse IoT environments.

In [27] the work proposed (CWVAEGAN-1DNN). This model consists of an encoder, decoder, and discriminator, all enhanced with one-dimensional convolutional layers. This architecture improves the model's capability to obtain complex patterns in the data. The model is trained to produce new instances of minority class data, which helps to rebalance the dataset. The 1D-CNN is utilized to categorize the data into normal and attack labels, leveraging the enhanced representation of the minority classes. However, its performance on the UNSW-NB15 dataset was slightly inferior compared to some advanced methods. This indicates that the model may not generalize as effectively across all datasets, suggesting a need for further optimization.

The study in [28] is experimented with Generative Adversarial Networks (BEGAN), to produce synthetic data for training the NIDS, build Autoencoders (AE) capable of offering dimensionality reduction, feature extraction, and detection models (DNN, CNN, and LSTM). The proposed system achieved 87% and 93% accuracy on UNSW-NB15 and NSL-KDD, respectively. Although the proposed framework has demonstrated enhancements in classification performance, it still showed rather poor detection rates for specific classes of threats. Table I shows the summary of related work.

Table I. SUMMARY OF RELATED WORK

Study	**Author and Year**	**Method**	**Key Findings**	**limitation**
[21]	S. Rahman, S. Pal, S. Mittal, T. Chawla, and C. Karmakar, 2024	Distributed GAN Network	Generates 10,000 synthetic samples per class (normal and attack), achieving competitive results in training ML models. Synthetic data quality was evaluated via boxplots to resemble real-world distributions.	Possible performance discrepancies in real-world applications due to exclusive training on synthetic data.
[22]	I. Ullah and Q. H. Mahmoud, 2021	cGAN, ocGAN, bcGAN	Balances and augments data for anomaly detection in IoT networks; bcGAN shows promising results for multiclass classification and improved anomaly detection.	Lower detection rates for sample sizes under 1000; improvements are needed for smaller datasets.
[23]	N. Abdalgawad, A. Sajun, Y. Kaddoura, I. A. Zualkernan, and F. Aloul, 2022	BiGAN, AAE	Produces synthetic data for IDS, achieving high F1-scores (up to 0.99) on IoT-23, outperforming traditional ML methods in cyberattack detection.	Lack of validation on non-IoT datasets; potential overfitting if dataset quality or diversity is low.
[24]	B. A. Alabsi, M. Anbar, and S. D. A. Rihan, 2023	CTGAN	Effectively generates synthetic data for IDS in IoT, improving detection of DoS and DDoS attacks and enhancing ML model training accuracy.	Difficulty adapting to novel attack types; limited applicability to diverse network environments.

[25]	J. He, X. Wang, Y. Song, Q. Xiang, and C. Chen, 2023	CWVAEGAN-1DNN	Uses encoder-decoder-discriminator architecture with 1D CNN layers to rebalance data by generating minority class samples, enhancing pattern recognition.	Slightly inferior performance on UNSW-NB15 dataset, indicating potential generalization issues.
[26]	C. Park, J. Lee, Y. Kim, J. G. Park, H. Kim, and D. Hong, 2023	BEGAN, AE with DNN, CNN, LSTM	Generates synthetic data and applies dimensionality reduction, achieving 87% accuracy on UNSW-NB15 and 93% on NSL-KDD, with improved classification performance.	Lower detection rates for specific threat categories; limited efficacy for certain types of threats

4. METHODOLOGY

Intrusion detection systems are crucial for observing network activities and detecting malicious behaviors. To handle the challenge of imbalanced datasets in IDS, [29] GANs were introduced as influential gadgets for producing synthetic data in various fields. CGANs extend this capability by generating data conditioned on labels, which can be specifically useful for classification tasks. [30] The principal difficulty addressed in this paper is the imbalance in network traffic datasets. The objective is to generate synthetic normal and attack data to balance the dataset and enhance the training of NIDS models. This paper leverages a Wasserstein GAN (WGAN) framework with conditional inputs (CGAN) to create artificial tabular data for binary and multi-labeled datasets.

Real data derived from network traffic was employed to generate artificial data instances, using comprehensive benchmark datasets for IDS, namely UNSW-NB15 and KDD CUP99. [31] These instances serve to optimize the accuracy of the intrusion detection model by ensuring an even distribution of the data. Utilizing Wasserstein Conditional Generative Adversarial Networks (WCGANs) facilitates the creation of a dataset that exhibits a balanced representation of all classes, necessitating the development of synthetic data for categories labeled as attacks in conjunction with those labeled as normal.

4.1 Dataset Preprocessing

Preprocessing the dataset by identifying features for the WCGAN models. Several subprocesses include handling missing values, outlier detection and removal, feature selection and filtering. Extreme outliers were identified and removed to improve the dataset's quality. This is done using the Interquartile Range (IQR) method, which detects points far outside the normal range of the data. Specifically, calculate the 10th (Q1) and 90th (Q3) percentiles of the numerical columns and remove records that were beyond three times the IQR:

$$IQR = Q1 - Q3 \qquad (3)$$

using this, establishing a lower bound at:

$$Q1 - 3 \times IQR \qquad (4)$$

and an upper bound at:

$$Q3 + 3 \times IQR \qquad (5)$$

which are designed to capture extreme outliers. Data points falling below or above these thresholds were considered outliers. This step reduced the UNSW-NB15 dataset from 257,673 records to 176,606 and the KDD CUP99 dataset from 148,517 to 103,613 rows, making the data more robust and less prone to overfitting when training neural network models.

The unique values and frequencies of the key categorical columns were calculated. Important categorical categories were filtered, and only the most relevant categories were kept. Less frequent categories were filtered out to focus on

the most impactful traffic types, like protocol, service and state in UNSW-N15 dataset and protocol, services and flags in KDD CUP99 dataset.

Random Forest is an ensemble learning approach that integrates numerous decision trees to attain enhanced accuracy and stability in predictions [32]. A Random Forest Classifier is employed to select a feature by figuring out each feature's significance in predicting the target label.

Training the model and extracting feature significance scores identify the characteristics that are most influential in distinguishing classes within the dataset. Recognizing these critical features to concentrate on the most pertinent variables improves the model's learning efficiency and diminishes the input space's dimensionality.

After performing outlier removal and feature selection, the subprocess is checked for missing values in the implemented datasets. Fortunately, the dataset did not contain any missing values, so no imputation or further data cleaning is required in this step. Categorical variables are also encoded utilizing one-hot encoding. This process converts each category into binary columns (0 or 1). The numerical features were standardized utilizing the StandardScaler:

$$x = \frac{x - \mu}{\sigma} \qquad (6)$$

This step ensures that all numerical attributes have a standard deviation $(\alpha)$of one and a mean (μ) of zero, prohibiting attributes with larger domains from dominating the model training process. [33]

Certain features in the dataset are binary, such these binary columns are identified and ensure that their synthetic versions remain binary. Some columns have only one unique value across the dataset. These columns are treated as constants, ensuring their values remain fixed when generating synthetic data. The final preprocessed UNSW-NB15 dataset contained 53 features and 157,035 records, and the KDD CUP99 dataset contained 78,543 rows and 35 features. The datasets were saved in CSV format separately for future use in training deep learning models for NIDS.

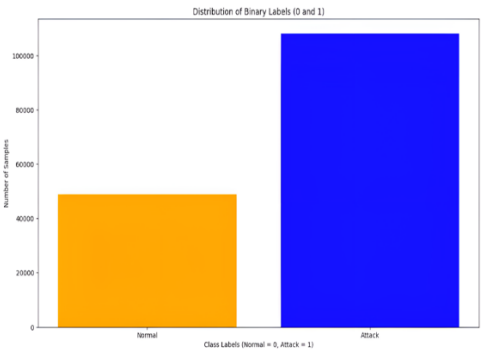

Fig. 4. Distribution of Binary Labels in UNSW-NB15 dataset

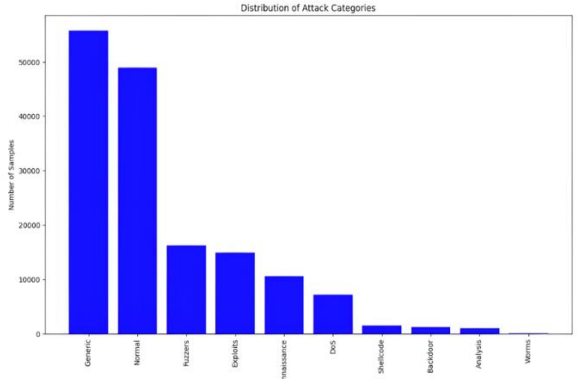

Fig. 5. Distribution of attack labels in UNSW-NB15 dataset

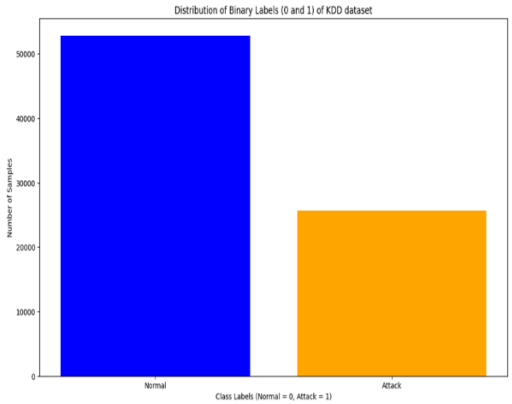

Fig.6. Distribution of Binary Labels in KDD CUP99 dataset

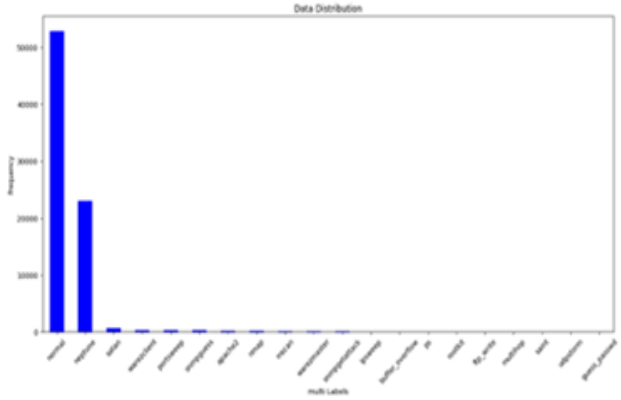

Fig.7. Distribution of attack labels in KDD CUP99 dataset

4.2 WCGAN Implementation

The WCGAN design consists of two primary components: a Generator and a Discriminator. The generator creates artificial data examples from random noise, while the discriminator differentiates between synthetic and real data. Both networks are trained in an adversarial process where the generator attempts to deceive the discriminator into classifying artificial samples as real. During generator training, the discriminator's weights are frozen [34]. For each batch of real data, a corresponding batch of synthetic data is produced by the generator. The discriminator is then trained to distinguish between synthetic and real data [35].

Two WCGAN models were implemented to address the unique requirements of binary and multi-class classification. The binary classification model generates synthetic data conditioned on binary labels (normal vs. attack). The multi-class model addresses the challenge of capturing the nuances of multiple attack categories within a single model.

The generator network employs fully connected dense layers with the LeakyReLU activations function. It takes as input a random noise vector and the corresponding class label vector, which guides the generator in the generation process. In the binary classification model, the output layer utilizes a sigmoid activation function [36] to produce synthetic data in the range [0, 1]. The discriminator network takes as input both real and synthetic data instances, along with their corresponding labels, and processes them through fully connected dense layers with the LeakyReLU activation function. Binary cross-entropy loss is employed for the binary classification model.

In the multiclass classification model, the generator employs an embedded layer to map categorical label input to a dense vector of size (latent_dim). The flattened layer converts the embedded label into a 1D vector. The concatenation step merges this label vector with noise as input. The discriminator also embeds and flattens the input to differentiate between real and fake data conditioned with specific labels. WCGAN framework utilizes the Wasserstein loss to stabilize training and mitigate mode collapse. MSE loss was utilized to measure the difference between the discriminator's prediction and the target value

To address class imbalance, the number of synthetic instances generated for each label is adjusted based on the distribution of the real dataset. This aims to balance the dataset, although the actual balance achieved may depend on the generator's ability to generate samples accurately for each class. Once trained, the generator produces synthetic data by feeding noise and labels as input. Post-processing steps are applied to ensure the validity of binary and single-value features in the synthetic data.

4.3 Synthetic Data Generation

The WCGAN was trained for 1000 epochs, after training, the generator produced 100,000 synthetic samples, which were then merged with the original dataset for further analysis. The synthetic data combined with the real data, created a more balanced dataset. The combined synthetic and real dataset is evaluated to ensure the label distribution and features are balanced. The distribution in the real, synthetic, and combined datasets was compared using various

methods. Finally, the data was shuffled and stored as a CSV file to utilize in training intrusion detection systems. Figure 6 illustrates the proposed WCGAN model.

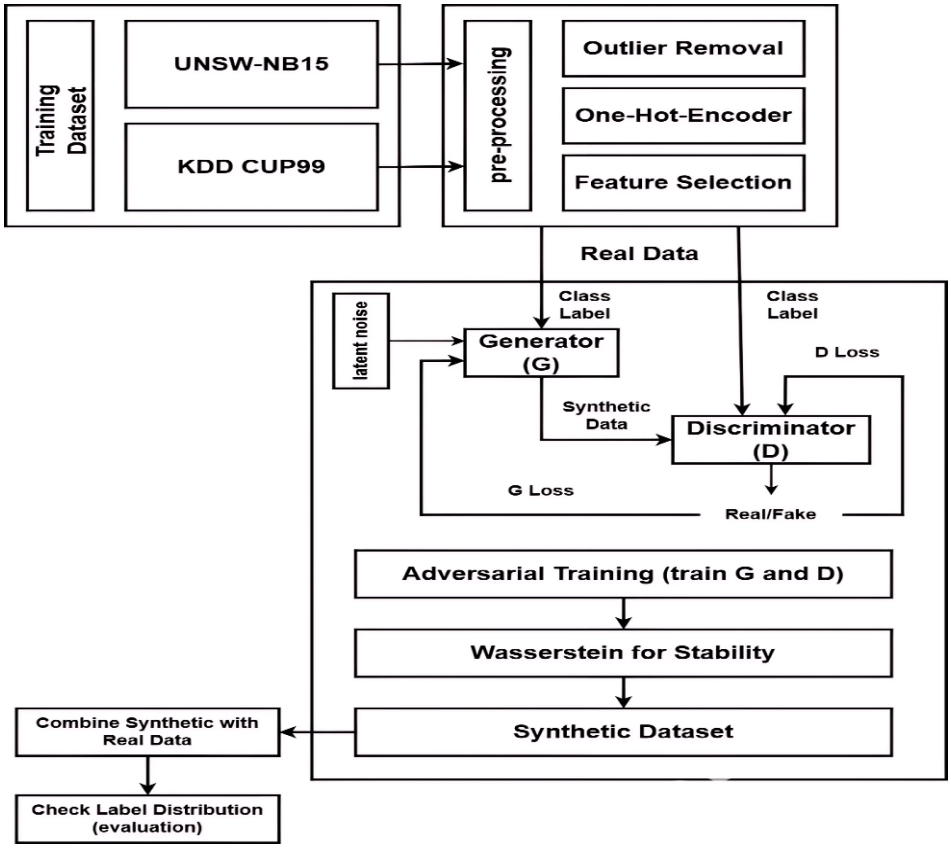

Fig. 8. The proposed WCGAN model

5. EXPERIMENTAL

The experiment was implemented by Python 3.12 using the 11th Gen, Intel(R) Core(TM) i5-1135G7 processor, which runs at 2.40GHz, and 8.00 GB of installed RAM.

4.4 UNSW-NB15 Dataset Description

The UNSW-NB15 is a frequently utilized dataset for evaluating IDS. It contains various attributes that are crucial for training AI-based models to identify network intrusions. UNSW-NB15 is developed by configuring the artificial environment at the UNSW cybersecurity laboratory. The IXIA technology employed has enabled the generation of a contemporary representation of both typical real-world network traffic and synthetic anomalous network traffic within a simulated ecosystem. UNSW-NB15 encompasses 9 principal categories of assaults through the utilization of the IXIA PerfectStorm technology. 49 attributes have been constructed utilizing Argus, Bro-IDS tools, and 12 algorithms that encompass the attributes of network packets. The UNSW-NB15 dataset distribution is shown in Table II.

Table II. THE DISTRIBUTION OF UNSW-NB15 DATESET

Classes	**Training dataset**	**Testing dataset**
Normal	56,000	37,000
Fuzzer	18,184	6,062
Generic	40,000	18,871
DoS	12,264	4,089
Backdoor	1,746	583
Exploit	33,393	11,132

Analysis	2,000	677
Shellcode	1,133	378
Reconnaissance	10,491	3,496
Worm	130	44

The UNSW-NB15 dataset comprises two subsets: UNSW-NB15 training set with 175,341 records and UNSW-NB15 testing set with 82,332 records. Every row comprises 43 features that denote network flow characteristics and two category features. One class feature signifies whether the record represents regular traffic (binary-valued attribute), While the other denotes the type of attack in cases of anomalous records.

Nine separate attack profiles are categorized as Analysis, Fuzzers, Backdoors, Generic, DoS, Reconnaissance, Exploits, Shellcode, and Worms. Attack types in UNSW-NB15 dataset are shown in Table III [37].

Table III. ATTACK KINDS IN UNSW-NB15 DATASET

Type of Attack	**Definition**
Fuzzer	The goal of this type of assault is to cause a system, application, or network to fail by flooding it with substantial quantities of arbitrary data.
Analysis	Intruders use ports to access web applications through web scripts and emails.
Backdoor	It is a method for discovering plaintext input while executing actions like circumventing secret authentication, gaining illegal access remotely to a device, and remaining undetected.
DoS	It is an assault that compromises computer memory resources, resulting in an enormous workload that obstructs approved requests from reaching the device.
Exploit	A series of commands that take benefit of a flaw, error, or weakness to induce unintended activity on a network or host.
Generic	It is an approach designed to induce a collision in any block cipher, irrespective of its setup.
Reconnaissance	It collects data on a device network to circumvent security measures.
Shellcode	A virus that injects into a small segment of code, commencing from a shell, to manipulate the hacked device.
Worm	An assault wherein the perpetrator duplicates and disseminates to further devices, frequently utilizing the device network for propagation, contingent upon the network traffic of the targeted computer employed for access.

KDD CUP99 Dataset Description

In the year 1999, the Massachusetts Institute of Technology created an intrusion detection dataset known as KDD CUP99. This dataset was generated by processing data from the Defense Advanced Research Projects Agency (DARPA) through several preprocessing stages, resulting in a comprehensive collection of records. KDD CUP99 comprises 41 distinct features, which are categorized into four groups: Host-based traffic features, Basic features, Content features, and Time-based traffic features.

The category identified as normal comprises 972,781 samples, while the category labeled as an attack includes 3,925,650 samples. The attacks are classified into four distinct categories, as detailed in Table IV. [38]

Table IV. ATTACK CLASSES OF KDD CUP99 DATASET

Attack Type	Description
User to Root Attack (U2R)	A perpetrator obtains root access to a system by taking advantage of a vulnerability present in a standard user account.
Remote to Local Attack (R2L)	The actor is capable of transmitting packets to a device; however, due to the absence of an account on that device, they exploit a weakness to gain local access as an authorized user of the device.
Danial of Service Attack (DoS)	An assault transpires when an individual excessively utilizes a device's memory in response to valid requests or obstructs access to the device for authorized users.
Probing Attack	The aim is to gather information regarding a network of devices with the intention of possibly circumventing its security measures.

4.5 Training Process of WCGAN for Binary Classification

Model training initiates with the generator producing synthetic data based on random noise and conditioned on the class label, with the aim of closely mimicking the real dataset. In binary classification, the labels take values of either 0 or 1.

The authenticity of the generated outputs against original samples is evaluated by the discriminator which is trained utilizing Adam optimizer and binary cross-entropy loss with a learning rate (α) of 0.00005. This adversarial process relies heavily on gradient descent optimization, which updates both networks iteratively to improve performance. The training is reiterated for 1000 epochs, and with a batch size of 64.

The discriminator is trained on real data along with their actual label, the target output for these samples is established at 1, representing real data. This phase adjusts the discriminator's weights to enhance its classification precision on authentic samples. The discriminator is subsequently trained using the synthetic data produced by the generator, together with the corresponding class labels. The target output for these samples is established at 0, signifying fake data. This phase promotes the enhancement of the discriminator in differentiating between produced samples and authentic samples.

To balance binary labels, the WCGAN model for binary classification is used to produce synthetic data. Specifically, the number of samples distributed to ensure equality in both labels. So, 79,637 new samples were created for the normal label (0), 20,363 new samples for the attack label (1) in the UNSW-NB15 dataset, as well as 36,409 new samples, were created for the normal label (0), and 63,591 new samples for attack label (1) in the KDD CUP99 dataset.

4.6 Training Process of WCGAN for Multi-class

WCGAN for multi-classification, built a generator and discriminator, embedding multiclass labels into the model. The training process encompasses interchanging among updating the discriminator and generator models to achieve a balanced game. The generator creates data correlated with the class label, and the discriminator is properly updated to classify the original and synthetic samples based on the class label.

The generator is instructed to create synthetic data that can mislead the discriminator. The RMSprop optimizer is utilized for training, which is standard for Wasserstein (GAN) training to maintain stable gradients, with a learning rate (α) of 0.00005 and iterating the training procedure for 1000 epochs, and a batch size of 64. In the multi-label classification task, the original dataset had (10) labels in UNSW-NB15, and (11) labels in KDD CUP99 datasets, each depicting different categories of network traffic (normal and various types of attacks). The distribution of these labels was highly imbalanced, which could lead to biased learning and poor model performance in distinguishing between attack categories. The artificial data generated by the WCGAN was merged with the original dataset to address class imbalances. The efficacy of this approach is evaluated by comparing the class distributions before and after the augmentation. The original dataset severely underrepresented certain attack categories in each dataset.

6. RESULT AND DISCUSSION

The preprocessing phase of the methodology involved identifying and removing outliers, encoding categorical features, addressing missing values, and scaling numerical data. After conducting these preprocessing steps, Random Forest was utilized to evaluate whether these modifications led to an effective feature representation for the classification task. Upon training the Random Forest Classifier, it assesses feature significance. The most significant aspects are emphasized, which can aid in model interpretation and feature selection.

The robustness of Random Forests makes them a suitable candidate for this purpose, as they can accommodate various data types and do not require significant feature scaling or normalization, offering valuable insights into the utility of the preprocessing steps.

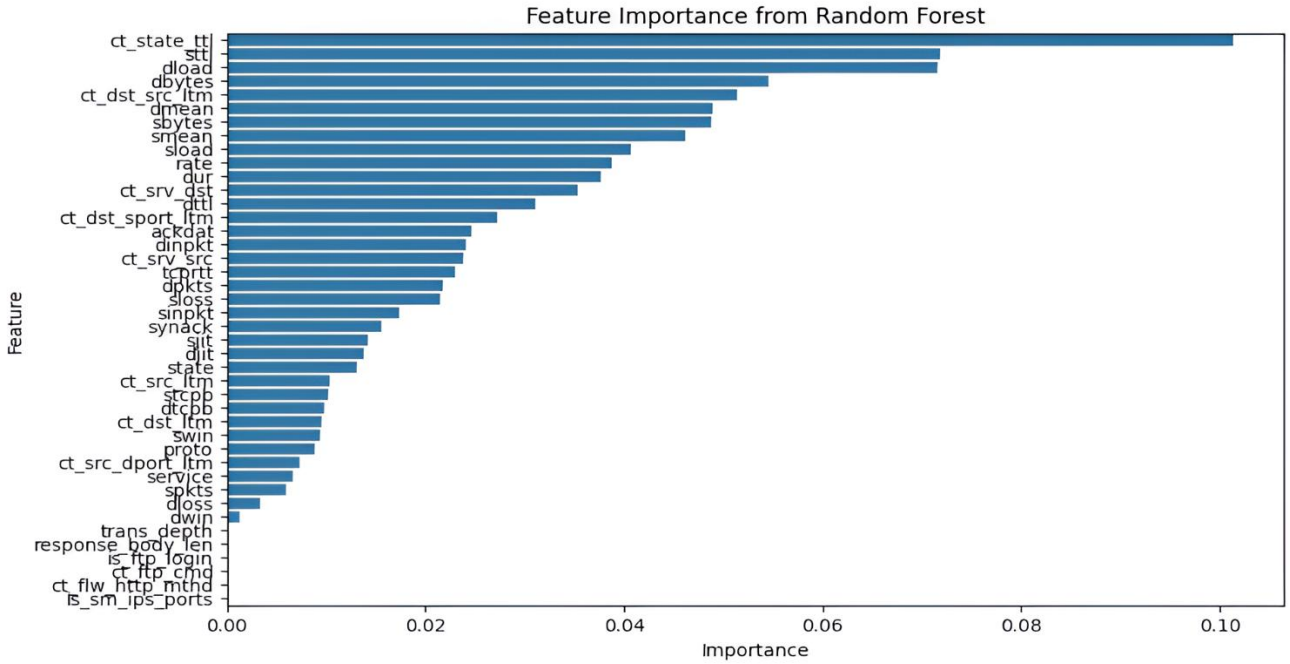

Fig. 9. Feature importance in UNSW-NB15 dataset

After merging the original data with the created synthetic data, the UNSW-NB15 dataset became perfectly balanced with 128,517 samples for the normal label (0), and 128,517 samples for the attack label (1), whereas the KDD CUP99 dataset balanced with 89,271 samples for the normal label (0), and 89,271samples for attack label (1). This balanced distribution allows the binary classification model to learn equally from both classes, reducing bias towards any particular label.

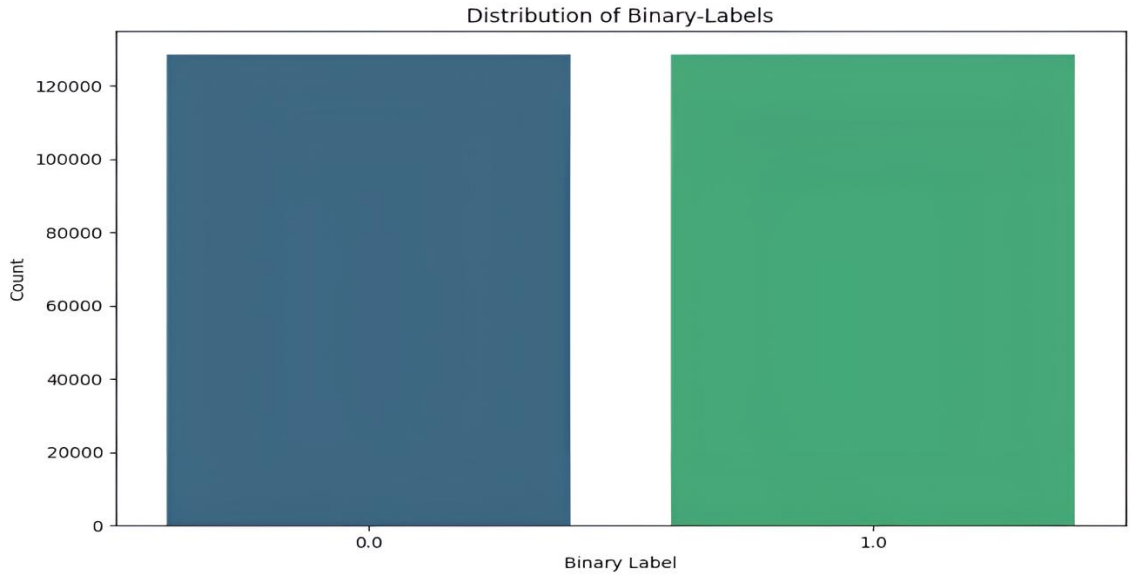

Fig. 10. Binary label distribution in UNSW-NB15 dataset

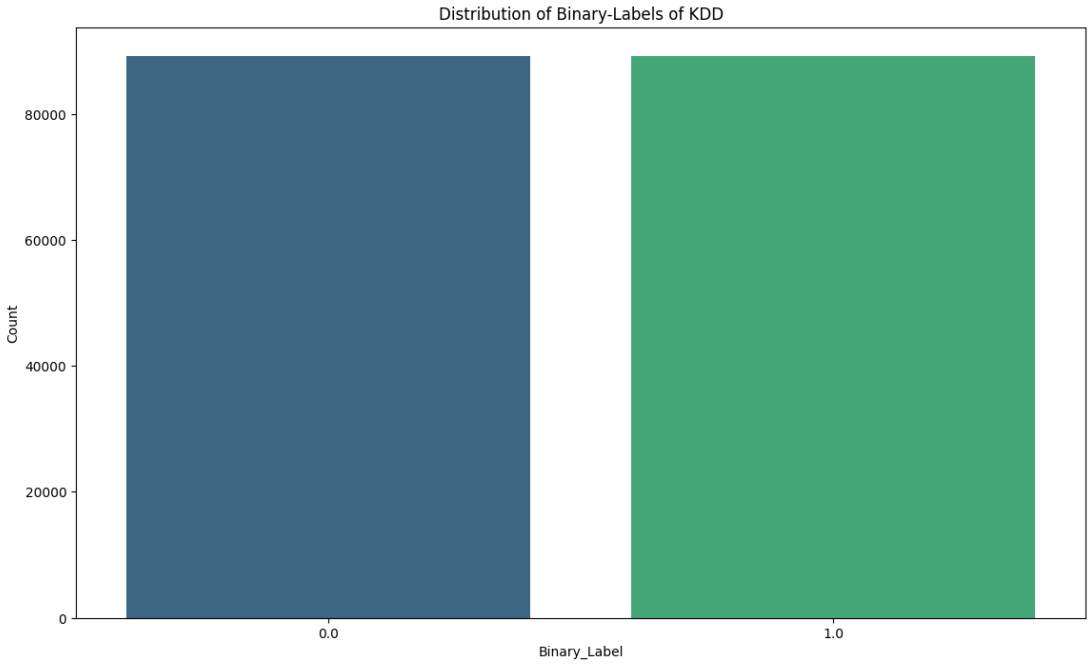

Fig. 11. Binary Distribution in KDD CUP99 Dataset

The goal was to increase the number of samples for underrepresented labels, creating a more balanced dataset for the multi-class classification mission. The WCGAN was specifically designed to generate samples for each label and maintain the relationships between the attack categories. After combining the original data with the synthetic data, the dataset became fully balanced, with each label having approximately 55,774 samples in the UNSW-NB15 dataset, whereas in the KDD CUP99 dataset, each label has approximately 52,862 samples. Performance metrics, such as discriminator accuracy, generator loss, and discriminator loss, are tracked over several epochs.

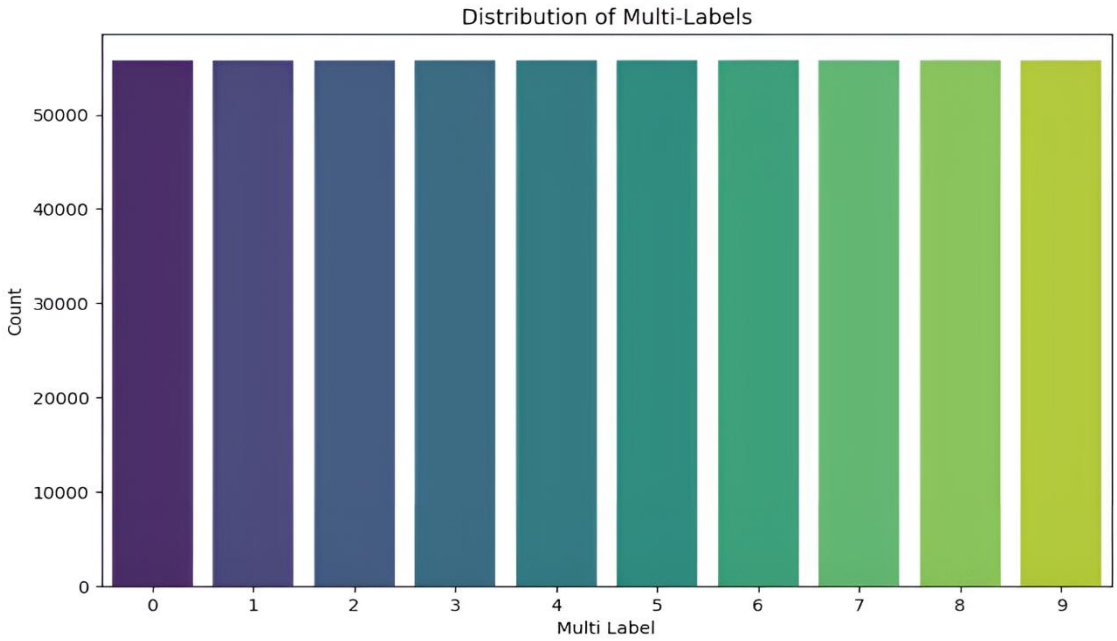

Fig. 12. Multi-Label distribution in UNSW-NB15 Dataset

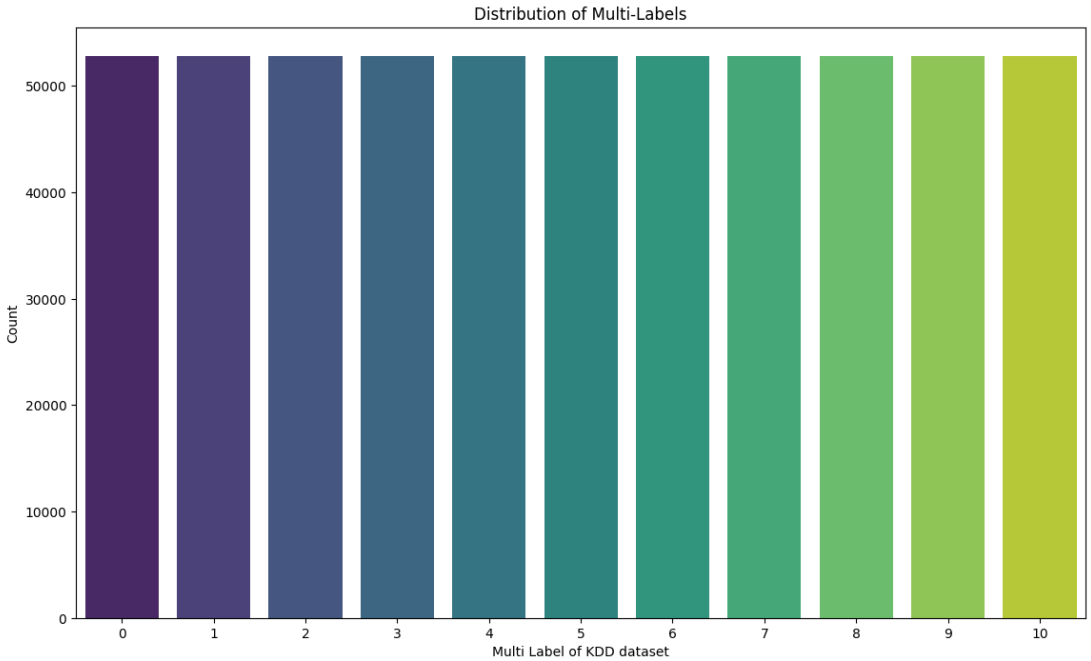

Fig. 13. Multi-Label Distribution in KDD CUP99 Dataset

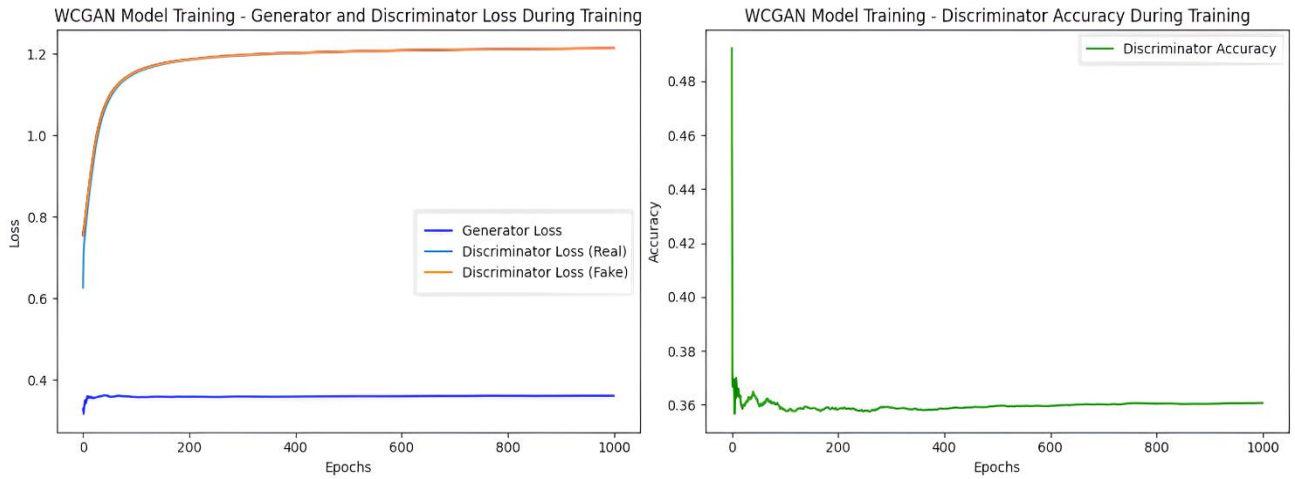

Fig. 14. WCGAN model training

The results demonstrate that the WCGAN effectively generates high-quality synthetic data, which helps balance the dataset. Generator Loss decreased steadily over training, indicating that the generator improved its capability to make realistic data. The discriminator's accuracy fluctuated during training but then stabilized, indicating that the discriminator learned to distinguish between real and synthetic data effectively.

Compared with distribution-based GAN, the WCGAN methodology uses a Wasserstein loss to evaluate how well artificial data aligns with the real data distribution by mathematically measuring the distance between the real data and the generated data. Although boxplots are a valuable supplementary visual tool for examining data distribution, they do not possess the quantitative rigor and feedback functionalities inherent in Wasserstein loss.

In CTGAN, The condition addresses the issue of imbalanced tabular data, thereby ensuring that the generator acquires the capability to learn and produce realistic samples across all feature combinations. CTGAN is specifically developed for the generation of tabular data, emphasizing the precise representation of both categorical and continuous variables within the generated datasets. In contrast, WCGAN focuses on generating samples for specific labels or categories; the condition helps balance datasets, such as normal traffic or particular types of attacks in an intrusion detection system. The produced data is specifically designed by the designated class labels, thereby ensuring that the distribution of the artificial data accurately corresponds to the distribution of actual data associated with those labels.

The proposed WCGAN-based method successfully cures the class imbalance trouble in NIDS datasets. By generating realistic attack data, the model allows for better training of deep learning-based IDS models, improving their ability to detect rare network attacks. This approach provides solutions to other binary and multiclass classification problems where data imbalance hinders the model's effectiveness.

Current supervised learning classifiers rely on the training set for their learning process, causing a significant drop in performance when facing new data types. Revealing these inherent data characteristics could result in a more reliable system that reduces the occurrence of false positives and negatives. As a result, GANs have been utilized to understand and depict the internal data distribution, using a training set that contains the network information of the data for classification purposes. This approach can be extended to other binary and multiclass classification problems where data imbalance is an issue.

7. CONCLUSION

This paper presents an approach utilizing WCGAN to produce synthetic data for network traffic dataset balance. The methodology improves the performance of NIDS models by enhancing their ability to detect rare attack traffic. One of the primary findings indicates that incorporating Wasserstein loss significantly enhances the stability of training compared to traditional GAN architectures, thereby mitigating common issues such as mode collapse. The generator benefits from Wasserstein loss by being able to learn from the distribution of the original dataset, and thus diversifying the random inputs, allowing the generator to produce varied samples. The WCGAN-based framework addresses class imbalance in intrusion detection datasets, improving system performance. It also generates synthetic data samples for

binary and multiclass scenarios, enabling more efficient training and improved accuracy in deep learning models. Future research could explore optimizing their architectural parameters and adapting the model to other domains, including video generation and reinforcement learning, to fully realize their potential. Additionally, understanding the interplay between conditioning information and generator performance could further refine WCGAN methodologies, fostering the development of even more robust generative models.

References

[1] A. D. Khaleefah and H. M. Al-Mashhadi, "Detection of IoT Botnet Cyber Attacks Using Machine Learning," *Informatica (Slovenia)*, vol. 47, no. 6, pp. 55–64, May 2023, doi: 10.31449/INF.V47I6.4668.

[2] I. Zografopoulos, J. Ospina, X. Liu, and C. Konstantinou, "Cyber-Physical Energy Systems Security: Threat Modeling, Risk Assessment, Resources, Metrics, and Case Studies," *IEEE Access*, vol. 9, pp. 29775–29818, 2021, doi: 10.1109/ACCESS.2021.3058403.

[3] T. Sowmya and E. A. Mary Anita, "A comprehensive review of AI based intrusion detection system," *Measurement: Sensors*, vol. 28, p. 100827, 2023, doi: https://doi.org/10.1016/j.measen.2023.100827.

[4] "Big Data Aggregation, Visualization and Clustering for Smart Grid in Smart City using Machine Learning," *Iraqi Journal of Computer, Communication, Control and System Engineering*, pp. 39–53, Sep. 2023, doi: 10.33103/uot.ijccce.23.3.4.

[5] M. A. Talukder *et al.*, "Machine learning-based network intrusion detection for big and imbalanced data using oversampling, stacking feature embedding and feature extraction," *J Big Data*, vol. 11, no. 1, Dec. 2024, doi: 10.1186/s40537-024-00886-w.

[6] K. Jiang, W. Wang, A. Wang, and H. Wu, "Network Intrusion Detection Combined Hybrid Sampling with Deep Hierarchical Network," *IEEE Access*, vol. 8, pp. 32464–32476, 2020, doi: 10.1109/ACCESS.2020.2973730.

[7] X. Zhou *et al.*, "Information Theoretic Learning-Enhanced Dual-Generative Adversarial Networks With Causal Representation for Robust OOD Generalization," *IEEE Trans Neural Netw Learn Syst*, 2023, doi: 10.1109/TNNLS.2023.3330864.

[8] Y. Cao, B. Sui, and W. Zhang, "REL-SAGAN: Relative Generation Adversarial Network Integrated With Attention Mechanism for Scene Data Augmentation of Remote Sensing," *IEEE J Sel Top Appl Earth Obs Remote Sens*, vol. 15, pp. 3107–3119, 2022, doi: 10.1109/JSTARS.2022.3166927.

[9] A. A. Al-Shargabi, J. F. Alshobaili, A. Alabdulatif, and N. Alrobah, "Covid-cgan: Efficient deep learning approach for covid-19 detection based on cxr images using conditional gans," *Applied Sciences (Switzerland)*, vol. 11, no. 16, Aug. 2021, doi: 10.3390/app11167174.

[10] A. Dunmore, J. Jang-Jaccard, F. Sabrina, and J. Kwak, "A Comprehensive Survey of Generative Adversarial Networks (GANs) in Cybersecurity Intrusion Detection," *IEEE Access*, vol. 11, pp. 76071–76094, 2023, doi: 10.1109/ACCESS.2023.3296707.

[11] M. Islam, G. Chen, and S. Jin, "An Overview of Neural Network," *American Journal of Neural Networks and Applications*, vol. 5, no. 1, p. 7, 2019, doi: 10.11648/j.ajnna.20190501.12.

[12] A. Shrestha and A. Mahmood, "Review of deep learning algorithms and architectures," 2019, *Institute of Electrical and Electronics Engineers Inc.* doi: 10.1109/ACCESS.2019.2912200.

[13] C. Qian, W. Yu, C. Lu, D. Griffith, and N. Golmie, "Toward Generative Adversarial Networks for the Industrial Internet of Things," *IEEE Internet Things J*, vol. 9, no. 19, pp. 19147–19159, 2022, doi: 10.1109/JIOT.2022.3163894.

[14] S. Y. Shin, Y. W. Kang, and Y. G. Kim, "Android-GAN: Defending against android pattern attacks using multi-modal generative network as anomaly detector," *Expert Syst Appl*, vol. 141, Mar. 2020, doi: 10.1016/j.eswa.2019.112964.

[15] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved Training of Wasserstein GANs," Mar. 2017, [Online]. Available: http://arxiv.org/abs/1704.00028

[16] W. Zhao, X. Chen, J. Chen, and Y. Qu, "Sample generation with self-attention generative adversarial adaptation network (SaGAAN) for hyperspectral image classification," *Remote Sens (Basel)*, vol. 12, no. 5, Mar. 2020, doi: 10.3390/rs12050843.

[17] S. W. Park, J. S. Ko, J. H. Huh, and J. C. Kim, "Review on generative adversarial networks: Focusing on computer vision and its applications," May 02, 2021, *MDPI AG*. doi: 10.3390/electronics10101216.

[18] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein Generative Adversarial Networks," in *Proceedings of the 34th International Conference on Machine Learning*, D. Precup and Y. W. Teh, Eds., in Proceedings of Machine Learning Research, vol. 70. PMLR, Oct. 2017, pp. 214–223. [Online]. Available: https://proceedings.mlr.press/v70/arjovsky17a.html

[19] Y. Xu, J. Yang, and M. Sawan, "Multichannel Synthetic Preictal EEG Signals to Enhance the Prediction of Epileptic Seizures," Apr. 2022, doi: 10.1109/TBME.2022.3171982.

[20] A. Zhang, L. Su, Y. Zhang, Y. Fu, L. Wu, and S. Liang, "EEG data augmentation for emotion recognition with a multiple generator conditional Wasserstein GAN," *Complex and Intelligent Systems*, vol. 8, no. 4, pp. 3059–3071, Aug. 2022, doi: 10.1007/s40747-021-00336-7.

[21] J. Engelmann and S. Lessmann, "Conditional Wasserstein GAN-based oversampling of tabular data for imbalanced learning," *Expert Syst Appl*, vol. 174, p. 114582, Jul. 2021, doi: 10.1016/J.ESWA.2021.114582.

[22] J. Wu *et al.*, "Sliced wasserstein generative models," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE Computer Society, Jun. 2019, pp. 3708–3717. doi: 10.1109/CVPR.2019.00383.

[23] S. Rahman, S. Pal, S. Mittal, T. Chawla, and C. Karmakar, "SYN-GAN: A robust intrusion detection system using GAN-based synthetic data for IoT security," *Internet of Things (Netherlands)*, vol. 26, Jul. 2024, doi: 10.1016/j.iot.2024.101212.

[24] I. Ullah and Q. H. Mahmoud, "A Framework for Anomaly Detection in IoT Networks Using Conditional Generative Adversarial Networks," *IEEE Access*, vol. 9, pp. 165907–165931, 2021, doi: 10.1109/ACCESS.2021.3132127.

[25] N. Abdalgawad, A. Sajun, Y. Kaddoura, I. A. Zualkernan, and F. Aloul, "Generative Deep Learning to Detect Cyberattacks for the IoT-23 Dataset," *IEEE Access*, vol. 10, pp. 6430–6441, 2022, doi: 10.1109/ACCESS.2021.3140015.

[26] B. A. Alabsi, M. Anbar, and S. D. A. Rihan, "Conditional Tabular Generative Adversarial Based Intrusion Detection System for Detecting Ddos and Dos Attacks on the Internet of Things Networks," *Sensors*, vol. 23, no. 12, Jun. 2023, doi: 10.3390/s23125644.

[27] J. He, X. Wang, Y. Song, Q. Xiang, and C. Chen, "Network intrusion detection based on conditional wasserstein variational autoencoder with generative adversarial network and one-dimensional convolutional neural networks," *Applied Intelligence*, vol. 53, no. 10, pp. 12416–12436, May 2023, doi: 10.1007/s10489-022-03995-2.

[28] C. Park, J. Lee, Y. Kim, J. G. Park, H. Kim, and D. Hong, "An Enhanced AI-Based Network Intrusion Detection System Using Generative Adversarial Networks," *IEEE Internet Things J*, vol. 10, no. 3, pp. 2330–2345, Feb. 2023, doi: 10.1109/JIOT.2022.3211346.

[29] A. S. Dina, A. B. Siddique, and D. Manivannan, "Effect of Balancing Data Using Synthetic Data on the Performance of Machine Learning Classifiers for Intrusion Detection in Computer Networks," *IEEE Access*, vol. 10, pp. 96731–96747, 2022, doi: 10.1109/ACCESS.2022.3205337.

[30] P. Isola, J. Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, Institute of Electrical and Electronics Engineers Inc., Nov. 2017, pp. 5967–5976. doi: 10.1109/CVPR.2017.632.

[31] Z. K. Maseer, R. Yusof, N. Bahaman, S. A. Mostafa, and C. F. M. Foozy, "Benchmarking of Machine Learning for Anomaly Based Intrusion Detection Systems in the CICIDS2017 Dataset," *IEEE Access*, vol. 9, pp. 22351–22370, 2021, doi: 10.1109/ACCESS.2021.3056614.

[32] A. Paul, D. P. Mukherjee, P. Das, A. Gangopadhyay, A. R. Chintha, and S. Kundu, "Improved Random Forest for Classification," *IEEE Transactions on Image Processing*, vol. 27, no. 8, pp. 4012–4024, Aug. 2018, doi: 10.1109/TIP.2018.2834830.

[33] I. N. Joudah and N. Abbas, "Asymptotically Unbiased Estimation of Mean and Standard Deviation in the Presence of Outlying Errors," *IEEE Access*, vol. 8, pp. 110623–110632, 2020, doi: 10.1109/ACCESS.2020.3002958.

[34] A. Dash, J. Ye, G. Wang, and H. Jin, "High Resolution Solar Image Generation Using Generative Adversarial Networks," *Annals of Data Science*, vol. 11, no. 5, pp. 1545–1561, 2024, doi: 10.1007/s40745-022-00436-2.

[35] M. Ozkan-Okay, O. Aslan, R. Eryigit, and R. Samet, "SABADT: Hybrid Intrusion Detection Approach for Cyber Attacks Identification in WLAN," *IEEE Access*, vol. 9, pp. 157639–157653, 2021, doi: 10.1109/ACCESS.2021.3129600.

[36] T. Sood, S. Prakash, S. Sharma, A. Singh, and H. Choubey, "Intrusion Detection System in Wireless Sensor Network Using Conditional Generative Adversarial Network," *Wirel Pers Commun*, vol. 126, no. 1, pp. 911–931, Sep. 2022, doi: 10.1007/s11277-022-09776-x.

[37] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *2015 Military Communications and Information Systems Conference, MilCIS 2015 - Proceedings*, Institute of Electrical and Electronics Engineers Inc., Dec. 2015. doi: 10.1109/MilCIS.2015.7348942.

[38] M. Tavallaee, E. Bagheri, W. Lu, and 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*, 2009, pp. 1–6. doi: 10.1109/CISDA.2009.5356528.