Knowledge-Based Systems 314 (2025) 113178

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

An enhanced BiGAN architecture for network intrusion detection

Mohammad Arafah [a],[*], Iain Phillips [b], Asma Adnane [b], Mohammad Alauthman [a], Nauman Aslam [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 Computer and Information Sciences, Northumbria University, Newcastle upon Tyne, United Kingdom

ARTICLE INFO

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

ABSTRACT

Intrusion detection systems face significant challenges in handling high-dimensional, large-scale, and imbalanced network traffic data. This paper proposes a new architecture combining a denoising autoencoder (AE) and a Wasserstein Generative Adversarial Network (WGAN) to address these challenges. The AE-WGAN model extracts high-representative features and generates realistic synthetic attacks, effectively resolving data imbalance and enhancing anomaly-based intrusion detection. Our extensive experiments on NSL-KDD and CICIDS-2017 datasets demonstrate superior performance, achieving 98% accuracy and 99% F1-score in binary classification, surpassing recent approaches by 7%–15%. In multiclass cases, the model achieves 89% precision for DoS attacks and 84% for Probe attacks, while maintaining 79% precision for rare U2R attacks. Time complexity analysis reveals 23% reduced training time while maintaining high-quality synthetic attack generation, contributing a robust framework capable of handling modern network traffic complexities and evolving cyber threats.

1. Introduction

Anomaly-based Intrusion Detection Systems (AIDS) are advanced cybersecurity solutions designed to detect abnormal or suspicious activity within a network or system. Modern AIDS systems are built using Artificial Intelligence (AI) and Machine Learning (ML) techniques to identify deviations from normal behavior, making them highly effective at detecting both known and unknown threats, including emerging attacks that lack predefined signatures.

Several studies have proposed robust AID systems that can detect a wide range of attacks [1,2] with ML algorithms to improve AID System (AIDS) performance. However, ML is deficient when learning against imbalanced datasets and massive features, as it is time-consuming and biased towards majority attacks in the training process [3,4].

Considering the aforementioned challenges associated with ML, researchers have explored deep learning (DL) approaches as a promising solution for AID systems, particularly in managing large-scale and complex network traffic data. While DL can offer advantages in learning complex patterns from large datasets, its effectiveness depends on various factors including data volume, computational resources, and data complexity. For sufficient training data and computational resources cases, DL approaches have demonstrated superior performance in detecting sophisticated attack patterns compared to traditional ML methods [5]. Generative models have emerged as effective tools to address the issue of imbalanced datasets in network attack traffic, a common challenge in AIDS. Specifically, Generative Adversarial Networks (GANs) and Bidirectional GANs (BiGANs) have been employed to synthesize high-quality attack samples that mimic real network traffic. By augmenting the dataset with these synthetic samples, GANs improve the balance between attack and benign data, enabling more accurate detection of diverse attack types and enhancing the overall performance of AID systems in complex network environments [6–8].

While the use of GANs has demonstrated significant potential in enhancing AID system performance, bidirectional architectures such as BiGANs further improve effectiveness by introducing an encoder network that maps data back to the latent space. This additional mapping helps the system capture richer feature representations. However, both GANs and BiGANs face limitations, including dependency on random input initialization, focus on abstract features that may overlook attack-specific details, and issues like stability challenges and mode collapse [9,10]. These factors can reduce their efficacy for detecting certain network attacks.

To address the challenges associated with existing GAN models, we propose E-BiGAN, a novel model based on parallel autoencoders (AE)

[*] 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), mohammad.alauthman@uop.edu.jo (M. Alauthman), nauman.aslam@northumbria.ac.uk (N. Aslam).

https://doi.org/10.1016/j.knosys.2025.113178
Received 19 December 2024; Received in revised form 5 February 2025; Accepted 11 February 2025
Available online 21 February 2025

and weight-sharing techniques. This innovative approach enhances the generation of high-quality synthetic attacks, which are then added to the original dataset. By augmenting the training data with these synthetic examples, the proposed model builds a more robust and effective AIDS, capable of detecting a wider range of network attacks and improving overall system performance. The E-BiGAN architecture comprises of three components: (1) a dual encoder design that enhances feature extraction across heterogeneous attack patterns, particularly for rare attack vectors, (2) a weight sharing mechanism between generators and discriminators that optimizes computational efficiency while preserving model stability, and (3) an enhanced loss function incorporating parallel AE components for generating high-quality synthetic attacks. These architectural advancements significantly improve upon traditional GAN-based methods, addressing data imbalance and reducing feature dimensionality while preserving detection accuracy across diverse attack patterns. The parallel training network architecture minimizes dependency between generators and discriminators, streamlining training complexity through shared parameters without sacrificing generation quality. A novel loss function based on parallel AEs and weight-sharing layers enhances feature learning. Additionally, dual encoding mechanisms enable superior capture of rare attack patterns, contributing to a robust and efficient framework for synthetic data generation and anomaly detection.

This paper is organized as follows: Section 2 provides a comprehensive review of related work in AIDS approaches, including both ML and DL methods. Section 3 introduces background concepts of GANs and BiGAN architectures. Section 4 describes the datasets used in training and testing. Section 5 demonstrates our proposed E-BiGAN model architecture and implementation details. Section 6 presents experimental results and comparative analysis. Section 7 discusses the findings and limitations of our approach. Finally, Section 8 concludes with key contributions and future work directions.

2. Related works

This section analyzes recent significant research approaches in Anomaly-based Intrusion Detection Systems (AIDS), focusing on machine learning, deep learning, and generative models approaches applied to AIDS. The analysis focuses on advances in machine learning, deep learning architectures, and generative models. We examine approaches addressing fundamental challenges in modern network security: high-dimensional data processing, imbalanced dataset management, and zero-day attack detection.

The landscape of network IDS has evolved significantly with the emergence of sophisticated cyber threats. Current approaches extend traditional signature-based methods, statistical anomaly detection, and advanced machine learning techniques. Recent developments in deep learning architectures have particularly transformed the field, with notable contributions in feature extraction [11], adversarial learning [12], and automated attack pattern recognition [13]. Within this context, generative adversarial networks have emerged as a promising paradigm for addressing data imbalance and synthetic attack generation, though challenges persist in stability and mode collapse [14].

2.1. Traditional machine learning approaches

Early machine learning approaches in IDS demonstrated varying effectiveness across different algorithmic architectures and feature selection methodologies. Decision tree-based classifiers showed particular promise, with Sahu and Mehtre's J48 implementation achieving 0.99 True Positive Rate (TPR) for known attack detection on the Kyoto dataset, though performance declined significantly to 0.10 TPR for unknown attacks [15]. This performance disparity highlighted a critical challenge in generalization capabilities for novel attack patterns.

Feature selection emerged as a crucial factor in improving detection accuracy. Gaikwad and Thool demonstrated this through a Genetic algorithm-based approach that successfully reduced the NSL-KDD feature space from 41 to 15 dimensions while maintaining detection efficacy [16]. Building on this foundation, Balogun and Jimoh proposed an innovative hybrid architecture combining Decision Trees with k-Nearest Neighbors (KNN) [17]. Their two-stage approach, validated on a 10% subset of the KDD Cup 99 dataset, achieved 99.7% accuracy, outperforming traditional methods including Naive Bayes and C4.5-based implementations.

More recent research has focused on ensemble methods and dimensionality reduction techniques. Ahmim et al. developed a sophisticated approach combining REP Tree and JRip algorithms with Forest PA, demonstrating improved binary classification capabilities [18]. Waskle et al. further advanced this direction by integrating Principal Component Analysis (PCA) with Random Forest classifiers, effectively addressing the computational overhead of large decision tree ensembles while maintaining competitive accuracy compared to SVM and Naive Bayes approaches [19]. Asif et al. [20] addressed the computational challenges of processing high-volume network traffic data through their MapReduce-Based Intelligent Model for Intrusion Detection (MR-IMID). Their architecture integrates machine learning techniques with distributed computing paradigms, enabling real-time processing of multi-source network data on commodity hardware. The framework implements an adaptive learning mechanism for unknown attack pattern detection, achieving 97.7% accuracy in training and 95.7% in validation phases.

Moving to the Random Forest (RF) classifier, Awotunde et al. [21] developed an advanced multi-level random forest architecture for intrusion detection in IoT networks, integrating correlation-based feature selection with genetic search algorithms and sequential forward selection. Their hybrid approach combines filter and wrapper methodologies for optimal feature subset identification, followed by fuzzy inference system classification to minimize misclassification errors. The framework demonstrated exceptional performance metrics, achieving 99.46% across accuracy, precision, sensitivity, and F1-score measurements, with 93.86% specificity.

Li et al. [22] conducted a comprehensive comparative analysis of feature reduction methodologies for machine learning-based intrusion detection systems in IoT networks. Their study empirically demonstrated that feature extraction approaches achieve superior detection performance metrics compared to feature selection methods when working with constrained feature sets, though at the cost of increased computational overhead. Their framework achieved optimal performance in both binary and multiclass classification scenarios, providing quantitative guidelines for implementing feature reduction strategies in IoT-specific intrusion detection systems based on operational requirements and computational constraints.

In a series of investigations, an SVM classifier was applied to obtain robust AIDS [23] addressed the escalating challenges of network security through a hybrid intrusion detection framework that integrates genetic algorithms with Support Vector Machine (SVM) classification. Their methodology implements feature optimization on the KDDCup99 dataset, successfully reducing the feature dimensionality from 42 to 29 dimensions while maintaining detection efficacy. The hybrid architecture demonstrated exceptional performance metrics, achieving 99.9% accuracy with a true positive rate of 0.99 and a false negative rate of 0.012. This integration of evolutionary computation with machine learning classification represents an effective approach to optimizing feature selection while maintaining high detection accuracy in network intrusion detection systems.

2.2. Deep learning architectures

The emergence of deep learning approaches has transformed intrusion detection capabilities, particularly in handling high-dimensional data. M. V. Yousefi-Azar et al. proposed an AE to learn from features input and reduce the dimensional data to preserve semantic learning for input features vector [24]. The proposed model was applied on the NSL-KDD dataset for malware classification and network-based AID. Also, AE played a significant role as a Generator in the learning process, where the AE learned by feature representation of AE architecture. The experimental results used four ML classifiers (NB, KNN, SVM, and Extreme Gradient Boosting), to measure the improvement in AIDS using the NSL-KDD dataset. The highest reported accuracy was 83.3 % by applying AE with the Gaussian NB classifier compared to 76.56 % with the use of all original features for the same classifier. However, it did not perform on the NSL-KDD official splits, which ensures that AIDS is evaluated under unseen attacks for training and testing sets. Indeed, random splitting for the NSL-KDD may not contain rare attacks. Consequently, it affects AIDS robustness and may not consider as a general solution in attacks detection.

Ieracitano et al. proposed a new approach, which depends on statistical analysis techniques for detecting outliers and AE for attack detection [25]. The proposed approach includes three steps; feature preprocessing, feature extraction, and attack classification. In the preprocessing step, a Median Absolute Deviation Estimator (MADE) method is applied to recognize the outliers and then apply min–max normalization to unify one scale $(0 - 1)$ for all features. After that, a one-hot encoder is applied to transform all textual features into numeric features. In the feature extraction and attack classification steps, AE was applied to extract the most correlated features to build AIDS using Multi-Layer Perceptron (MLP), Linear-Support Vector Machine (L-SVM), Quadratic-Support Vector Machine (Q-SVM), Linear Discriminant Analysis (LDA), Quadratic Discrimination Function (QDF) and the Long Short-Term Memory (LSTM) classifiers. The proposed approach was evaluated in binary and multiclass classification using the NSL-KDD dataset [25]. The results showed that the AIDS-based AE approach outperformed other classifiers in terms of Precision, Recall, and F1 score metrics, as indicated in Table 8.

In another study, a new model was proposed to combine the advantages of ML and DL by Mighan, S.N. and Kahani, M. [26]. Apache Spark was used for data processing which is capable of processing a vast amount of traffic compared to ML. After that, a stacked AE was applied as feature extraction. To detect attacks, the model applied several ML classifiers (RF, DT, NB) on UNB ISCX 2012 dataset to validate the results. Apache Spark and stacked AE architecture help select features accurately, even with large-scale data, leading to high performance in attack detection. However, only one dataset was used to validate the robustness of the proposed model.

Moving to VAE, a type of DL approach that represents the improved AE version, which has been applied in different directions, including AIDS. VAE proved that it could learn from features even if the data grows continuously [27], which was applied for the KDD Cup 1999 and MNIST datasets. The performance of AID VAE based is better than upper bound and Elastic Weight Consolidation (EWC) methods in terms of the Receiver Operating Curve (ROC) metric [27]. In another research work applied using VAE, it outperformed oversampling methods such as Synthetic Minority Oversampling Technique (SMOTE), Adaptive Synthetic (ADASYN), and generative models such as GANs in terms of standard metrics on MNIST and NIST19 datasets [28].

2.3. Recent hybrid approaches

GANs suffer from stability and produce high-quality attacks. Also, it can learn and generate from only one class of dataset, known as mode collapse. All these factors have directly affected attacks' quality and then AIDS performance [25,27]. In contrast, BiGAN suffers slightly from GANs challenges due to the BiGAN model concentrating on essential features. An AE and VAE were utilized to enhance AIDS to detect a wide range of attacks, which led to BiGAN model later [3].

Recently, W. Xu et al. [29] proposed using the BiGAN model to reduce the dependency between the Generator and Discriminator. The model included a one-class classifier (binary classification) using AE on NSL-KDD and CIC-DDoS2019 datasets [29]. Zhang evaluated BiGAN by tuning its parameters to obtain the best attack quality [11]. Sample sizes, training times (epochs), learning rate, and the number of hidden layers for the Generator and Discriminator parameters were investigated in each experiment. Referring to conducted experiments, the BiGAN performance was not improved by parameter tuning as no feature selection was applied. In another study applied BiGAN, Kaplan, M. Oguz, and S. Emre Alptekin proposed a new training method in BiGAN architecture for anomaly detection. They suggested training the Generator and Discriminator in two phases first separately and then training together [3]. This technique requires one additional reconstruction loss function to train the Generator until it learns enough to produce high-quality samples. After that, the Discriminator and Generator learn together. When the Generator and Discriminator had completed the learning stage and produced minimum loss scores, the BiGAN Discriminator was tested against unseen attacks [3], which delivered high results in attacks' detection.

In a most recent study, a new model was introduced to detect attacks based on RNN, LSTM, and GRU, which was applied on NSL-KDD and UNSW-NB15 datasets [30]. The model applied XGBoost to select the most relevant features ranging from 17 to 21. The highest test accuracy on the NSL-KDD and UNSW-NB15 datasets were 88.13 % and 87.07 %, respectively. In separate research, GANs and BiGAN were evaluated to produce new attacks from the NSL-KDD and CICIDS-2017 datasets. These models delivered limited enhancement for AIDS compared to other generative models based on performance metrics [31]. As a result, there is a need to enhance BiGAN to deliver efficient AIDS, which is resolved in our proposed model.

Recent advances in neural network architectures for security applications have demonstrated promising results in handling delayed and stochastic systems. Notable work by [32] on exponential state estimation for delayed competitive neural networks provides valuable insights for handling temporal aspects in network traffic analysis. Similarly, [33] contributed significant findings on stochastic Markovian jump systems, relevant to our treatment of network state transitions. The event-triggered approach proposed by [34] offers complementary perspectives on handling delayed networked systems, particularly pertinent to real-time intrusion detection scenarios.

Contemporary research has increasingly focused on hybrid architectures integrating multiple methodological approaches. Arafah et al. [14] present an innovative hybrid architecture integrating denoising autoencoders with Wasserstein Generative Adversarial Networks (WGAN) to address fundamental challenges in network intrusion detection systems. Their AE-WGAN framework effectively mitigates high-dimensional feature space complexity while simultaneously generating high-fidelity synthetic attack patterns to resolve class imbalance issues. Through comprehensive empirical validation on NSL-KDD and CICIDS-2017 datasets, the architecture demonstrates superior performance metrics across both binary and multiclass classification scenarios, particularly in detecting low-frequency attack patterns. The framework's efficacy is validated through extensive computational complexity analysis, establishing its viability for real-world deployment while maintaining robust detection capabilities against emerging cyber threats.

The related works explored several models and approaches that have been used to build AIDS from a learning perspective and an oversampling perspective (AE, VAE, GANs, BiGAN, AE-WGAN). AIDS based on the ML approach indicated limited performance across large-scale data with high-dimensional data compared with the DL approach. However, recent research related to AE and VAE, which led to the BiGAN model, has been covered in the context of intrusion detection. Despite their potential to improve performance significantly using DL approach, few researchers have used BiGANs for AIDS purposes, which mainly used image processing and computer vision context [35–37].

Table 1 presents a comparative analysis of recent approaches in network intrusion detection systems. The comparison focuses on key

Table 1
Comparison of related work in network intrusion detection systems.

Study	Method	Dataset	Features	Performance Metrics	Limitations
[38]	AE	NSL-KDD	41	Acc: 83.3%	Limited feature extraction
[39]	AE	NSL-KDD	41	F1: 75%	No rare attack handling
[40]	BiGAN	NSL-KDD	41	F1: 92%	Mode collapse issues
[41]	BiGAN	NSL-KDD	41	F1: 90%	Training instability

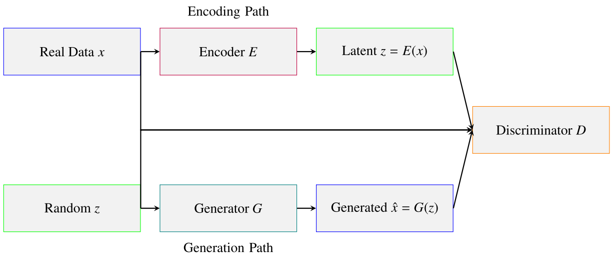

Fig. 1. Bidirectional generative adversarial network (BiGAN) architecture.

Table 2
NSL-KDD dataset distribution across category names [43].

Behavior types	Category name	Training set	Testing set
Anomaly	DoS	45,927	7458
	R2L	995	2754
	U2R	52	200
	Probing	11,656	2421
Normal		67,343	9711
Total		125,973	22,544

aspects including methodological approach, dataset utilization, feature dimensionality, performance metrics, and notable limitations. This analysis reveals common challenges in existing solutions, particularly in handling rare attacks and maintaining model stability, which our proposed approach aims to address.

3. Background

This section describes GANs and BiGAN architectures, their structures and learning processes.

3.1. Standard/vanilla GANs

Goodfellow introduced the GAN architecture [42], which generates new high-quality samples based on learning from the distribution of the given data. The Generator tries to fool the Discriminator while the Discriminator tries to identify fake samples. A GAN's training process starts by taking random noise as a vector, denoted Z. Z is sent to the Generator (G) to generate new samples, which creates a new distribution, denoted as P_G. The Discriminator receives these fake samples along with real samples. It computes the probability of an incoming sample being real ($P \approx 1$) or fake ($P \approx 0$).

The Generator and Discriminator work against each other in a minimax game [42]. We define Generator loss (G-loss) as the difference between generated and actual samples. Similarly, the Discriminator samples classified wrongly are termed Discriminator loss (D-loss).

3.2. Bidirectional GAN

A BiGAN is a type of GAN [35] that includes an Encoder (E) and a mapping function. The encoder transfers input features from high into lower dimensional space as latent vector representation. The Discriminator applies the classification process on attacks produced by the Generator and original attacks encoded by the encoder. Fig. 1 shows the architecture of the BiGAN [35].

The BiGAN model starts with real input data (x), where the inverse learning capability is applied by mapping the real input data into an encoded format ($E(x)$). Next, the Generator converts a random vector (z) into an adversarial sample format ($G(z)$). After that, the discriminator D distinguishes the joint distribution between fake and real samples.

3.3. Advantages of BiGAN architecture in intrusion detection

The BiGAN architecture presents significant methodological advantages for network intrusion detection through its innovative architectural design and operational characteristics. At its core, the bidirectional learning capability enables simultaneous optimization of encoding and generation processes, facilitating enhanced feature representation through inverse mapping and improved capture of complex attack signatures.

The architecture demonstrates superior stability characteristics compared to traditional GANs, evidenced by reduced mode collapse through encoder feedback mechanisms and optimized gradient flow during the training phase. This stability translates into robust feature learning capabilities, where the model constructs rich latent space representations of network traffic patterns, enabling fine-grained discrimination between normal and anomalous behaviors while maintaining strong generalization to previously unseen attack variants.

From an implementation perspective, the architecture achieves computational efficiency through strategic parameter sharing, effectively handling high-dimensional network traffic data while maintaining scalability for large-scale deployment scenarios. These advantages collectively position the BiGAN architecture as a particularly effective framework for addressing the complexities of modern network intrusion detection systems.

4. Datasets

In this section, we show statistical information for the well-known NSL-KDD and CICIDS-2017 datasets. The attack names, categories and counts are indicated to show the diversity in attack classes. These datasets are examples used to train network-based AIDS. However, the shortage of this type of dataset leads a lack of diversity in IDSs [44].

4.0.1. NSL-KDD

NSL-KDD, created by the Canadian Institute for Cybersecurity (CIC) in 2017, is an enhanced version of the KDD CUP99 dataset with no redundant records [43,45]. Table 2 shows the number of training and testing records sets for each attack category according to CIC split.

4.0.2. CICIDS-2017

CICIDS-2017 is dataset that has complete network configurations covering many different devices and operating systems. The dataset contains complete network traffic for 12 user profiles, with each record labeled either benign or attack. The interaction between Local Area Network (LAN) and other networks reflects a complete interaction and covers different protocols (HTTP, HTTPS, FTP, SSH).

Attack Diversity, Heterogeneity, Feature Set and mentioned characteristics make this dataset comprehensive and robust for training and evaluating AIDS. This dataset contains different attack categories: Brute Force, Heartbleed, Botnet, DoS, DDoS, Web, and Infiltration attacks, described in 80 features. These features are collected over five days. Table 3 shows the executed attacks in the CICIDS-2017 dataset for each day.

5. E-BiGAN model

The bidirectional structure of BiGAN shows different advantages for anomaly detection in network security applications. These advantages derive from three fundamental characteristics. First, the bidirectional architecture enables simultaneous learning of data generation and inverse mapping processes, enabling comprehensive feature representation. Second, the encoder network effectively captures the underlying

Table 3
The attacks detected on the CICIDS-2017 dataset per day.

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

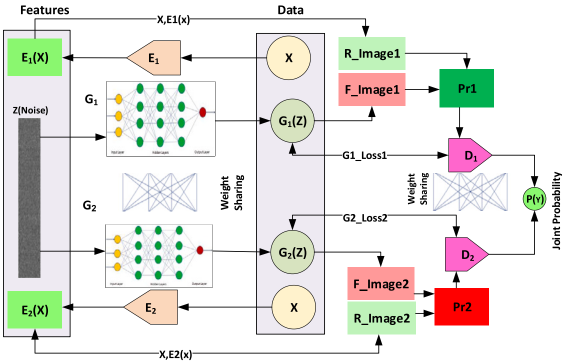

Fig. 2. E-BiGAN architecture.

distribution of normal network traffic patterns. Third, the learned latent space representation delivers an efficient mechanism for detecting behavioral deviations in network traffic.

Despite these advantages, traditional BiGAN architectures encounter significant limitations when generating samples for rare network attacks. The primary challenge is mode collapse, where the generator converges to producing a limited subset of samples instead learning the complete distribution of attack patterns. This limitation is particularly critical in IDS, where the capability to represent diverse attack patterns is essential. Furthermore, the single encoder architecture in traditional BiGAN may suffer from capturing the complex, multidimensional features inherent in network traffic data.

To address these limitations, we propose the Enhanced BiGAN (E-BiGAN) architecture with three principal innovations:

1. A dual encoder architecture that implements parallel feature extraction paths, enhancing the capture of diverse feature representations and mitigating mode collapse risk.
2. A weight-sharing mechanism between generators and discriminators that ensures consistent feature learning while optimizing model parameters.
3. An enhanced loss function that integrates parallel autoencoder components with weight-sharing elements, facilitating comprehensive attack pattern representation.

5.1. E-BiGAN definition

The architecture of the E-BiGAN model is based on the shareable weights of the BiGAN model. By adding a new encoder, Generator, and Discriminator to the BiGAN model, the quality of generated attacks has been directly impacted by the sharing weight layers used between Generators and discriminators. Fig. 2 shows the proposed architecture of the E-BiGAN.

The architecture takes one input from the real data then passes it into two encoders $(E1, E2)$ and produces two samples similar to the real data using two encoder processes: $(X, E1)$ and $(X, E2)$ respectively. The two samples produced by two encoders are encoded (compressed) into a latent vector. The difference between the real sample and the produced sample represents the loss.

Consequently, there are two losses for each encoder, as indicated in Fig. 2. Meanwhile, two noise vectors are selected and given to generators. The generated samples within Generator processes $(G1, Z), (G2, Z)$ are fake samples depending on random noise with a Gaussian distribution. Now, each Discriminator tries to find the probability of the sample being real or fake. In other words, the Discriminator tries to maximize the probability of a real compressed sample to original data (close to one), and the generated samples are similar to fake samples (close to zero).

The two generators $(G1, G2)$ in the E-BiGAN have a separate neural network in which the weights are shared in an independent shared neural network, as indicated in Fig. 2. The shared neural network between generators significantly impacts the production of high-quality samples [46]. These samples are obtained by two encoders for the same input, which focuses on the features used intensely. The shared network between the generators decreases the number of parameters, which decreases the training time compared with current approaches.

The loss functions of the E-BiGAN in each Generator has the same loss function in vanilla GANs. However, the objectives in E-BiGAN are different, which affects the minimax equation due to the new components added to this architecture. The first objective in E-BiGAN architecture is to minimize the Generator (Fake Images (F_Image1, F_Image2)) and encoder (Real Images (R_Image1, R_Image2)) loss. While the second objective is to maximize the Discriminator loss. The E-BiGAN objectives can be represented in Eqs. (1) and (2), respectively.

$$MinG_1(z, \theta_{G_1}), G_2(z, \theta_{G_2}), E_1(x, \theta_{E_1}), E_2(x, \theta_{E_2}) \tag{1}$$

$$MaxD_1(\{x, z\}; \theta_{D_1}), D_2(\{x, z\}; \theta_{D_2}) \tag{2}$$

The minimax equation in E-BiGAN contains six components: two discriminators $(D1, D2)$, two generators $(G1, G2)$, and two encoders $(E1, E2)$, which are different from the minimax game of GANs as in vanilla GANs, the Discriminator tries to discriminate between the real and adversarial samples, as in Eqs. (3) and (4), respectively.

$$D_{real} = D_1(\{x, E_1(x)\}; \theta_{D_1}) + D_2(\{x, E_2(x)\}; \theta_{D2}) \tag{3}$$

$$D_{fake} = (\{G_1(z; \theta_{G_1}), z\}; \theta_{D1}) + (\{G_2(z; \theta_{G_2}), z\}; \theta_{D2}) \tag{4}$$

According to the minimax game equation, the discriminators between real and fake samples $(D1$ and $D2)$ can be substituted in Eq. (5), as indicated in Eq. (6). The first term of Eq. (6) represents the discrimination of real samples, whereas the second term represents the discrimination of fake samples. It is important to mention that the θ_D represents the used weights parameter in the Discriminator networks $D1$ and $D2$. This parameter is updated in the backpropagation of the discrimination within the training process.

$$\begin{aligned}\min_{\theta_G}\max_{\theta_D} V(G, D) = \min_{G}\max_{D}\mathbb{E}_{x\sim P_{data}}[\log D(x)]+ \\ \mathbb{E}_{z\sim P_z}[\log(1 - D(G(z)))]\end{aligned} \tag{5}$$

$$\begin{aligned}V(D_1, D_2, G_1, G_2, E_1, E_2) = \log \\ (D_1[\{x, E_1(x, z_1)\}; \theta_{D_1})]+ \\ \log(D_2[\{x, E_2(x, z_2)\}; \theta_{D2})] + \\ log(1 - D_1[\{G_1(z_1; \theta_{G_1}), z_1\}; \theta_{D1}]) + \log \\ (1 - D_2[\{G_2(z_2; \theta_{G_2}), z_2\}; \theta_{D_2}])\end{aligned} \tag{6}$$

Eq. (6) is defined to work on a single sample input, but for multiple samples of traffic, the expectation (mean) is included, as follows in Eq. (7).

$$\begin{aligned}V(D_1, D_2, G_1, G_2, E_1, E_2) = \\ E_{x\sim p_x(x)}E_{z\sim p_E(z|x)}[\log(D_1[\{x, E_1(x, z_1)\}; \theta_{D_1})] \\ +\log(D_2[\{x, E_2(x, z_2)\}; \theta_{D2})] \\ ]+E_{z\sim p_z(z)}E_{x\sim p_G(x|z)}[\log(1 - D_1[\{G_1(z_1; \theta_{G_1}), z_1\}; \theta_{D1}])+ \\ \log(1 - D_2[\{G_2(z_2; \theta_{G_2}), z_2\}; \theta_{D_2}])]\end{aligned} \tag{7}$$

Table 4
The architecture of the E-BiGAN.

Parameters	NSL-KDD Dataset	CICIDS-2017 Dataset
Input Image (H× W)	4 × 6 × 1	2 × 5 × 1
Latent Dimension	32	32
Optimizer of the BiGAN	Adam (0.0009, 0.5)	Adam (0.0009, 0.5)
Number of Layers (Encoder)	7	7
Activation Function (Encoder)	ReLu	Linear
Number of Layers (Generator)	5(shared 4)	5(shared 4)
Number of Layers (Discriminator)	1(shared 4)	1(shared 4)
Activation Function (Discriminator)	ReLu	Linear
Epoch	500	1000

Eq. (7) works only for a discrete (discrete samples) distribution since the expectation is used. However, the expectation is replaced with integral to work in a continuous distribution as indicated in Eq. (8).

$$
\begin{aligned}
&V(D_1, D_2, G_1, G_2, E_1, E_2) = \\
&\int_x P_x(x) + \int_z P_E(z|x)[\log(D_1[\{x, E_1(x, z_1)\}; \theta_{D_1})]+ \\
&\log(D_2[\{x, E_2(x, z_2)\}; \theta_{D2})]+ \\
&\int_x P_x(x) + \int_z P_E(z|x)[\log(1 - D_1[\{G_1(z_1; \theta_{G_1}), z_1\}; \theta_{D1}])+ \\
&\log(1 - D_2\{G_2(z_2; \theta_{G_2}), z_2\}; \theta_{D_2}])
\end{aligned} \tag{8}
$$

Eq. (8) is simplified using a single integral to both terms and joint distribution property, as indicated in Eq. (9).

$$
\begin{aligned}
&V(D_1, D_2, G_1, G_2, E_1, E_2) = \\
&\int_{xz} P_{E_{12}x}(x, z)[\log(D_1[\{x, E_1(x, z_1)\}; \theta_{D_1})]+ \\
&\log(D_2[\{x, E_2(x, z_2)\}; \theta_{D2})]]+ \\
&\int_{xz} P_{G_{12}z}(x, z)[\log(1 - D_1[\{G_1(z_1; \theta_{G_1}), z_1\}; \theta_{D1}])+ \\
&\log(1 - D_2[\{G_2(z_2; \theta_{G_2}), z_2\}; \theta_{D_2}])]
\end{aligned} \tag{9}
$$

Applying the derivation on Eq. (9) produces the optimal case for discriminators $D1$ and $D2$. Moreover, the generators and discriminators introduced in vanilla GANs are optimized by shared network in E-BiGAN architecture, as indicated in Eq. (10).

$$
D * [\{x, z\}; \theta_D] = \frac{P_{Ex}(x, z)}{P_{Ex}(x, z) + P_{Gz}(x, z)} \tag{10}
$$

5.2. E-BiGAN architecture

The previous works on generative models can be summarized into three directions: network architecture, loss function and optimization algorithm [7]. This section describes the neural network architecture of E-BiGAN that is applied to build robust AIDS.

Firstly, encoder networks contain eight layers that take an input vector to encode it to the generator as an output vector. The used activation function is the *ReLu* in the NSL-KDD and the *linear function* in the CICIDS-2017 dataset. Additionally, the batch normalization function is used to optimize the encoder network.

The neural network architecture of the Generator includes four layers which take the encoded features (vector) as an input. The *ReLu* activation function is used with the batch normalization function to enhance the performance of the networks. It is essential to mention that the weights between two generators are implemented in four layers with two functions: *ReLu* activation and batch normalization. Consequently, the structure of both generators is identical, which helps the generators to learn rabidly.

Similarly, the shared networks in discriminators contain four layers with two functions: *ReLu* activation and batch normalization. For the Discriminator networks, it only contains a single layer. Table 4 shows the used E-BiGAN architecture in building AIDSs for NSL-KDD and CICIDS-2017 datasets.

To prove the enhancement achieved by the E-BiGAN model in AIDS impact, BiGAN was built to compare the performance of these models in order to demonstrate the improvement gained by E-BiGAN in AIDS impact. In addition, the E-BiGAN and BiGAN models have been utilized to fit two datasets: NSL-KDD and CICIDS-2017 in AIDS, these datasets are variants in nature, data distribution, and the number of attacks. So, the best parameters' values are determined for each model by experiments based on AIDS performance for each dataset. All experiments are conducted based on real and generated attacks obtained by E-BiGAN in both datasets (one model for each dataset) to obtain the highest scores. Table 5 shows the BiGAN architecture used in both datasets compared with the E-BiGAN model.

Table 5
The architecture of the BiGAN.

Parameters	NSL-KDD Dataset	CICIDS-2017 Dataset
Input Image (H× W)	4 × 6 × 1	2 × 5 × 1
Latent Dimension	32	32
Optimizer of the BiGAN	Adamax (0.0009, 0.9, 0.999)	Adamax (0.0009, 0.9, 0.999)
Number of Layers (Encoder)	6	6
Activation Function (Encoder)	ReLu	Linear
Number of Layers (Generator)	5	6
Number of Layers (Discriminator)	6	4
Activation Function (Discriminator)	ReLu	Linear
Epoch	500	1000

Table 6
Attacks quality using BiGAN and E-BiGAN models.

Metric/Dataset	NSL-KDD		CICIDS-2017	
	BiGAN	E-BiGAN	BiGAN	E-BiGAN
Cosine similarity	7.0081	**7.5526**	3.1044	**11.1114**
Discrimination score	0.7290	**0.6894**	0.4791	**0.4496**

5.3. Data-preprocessing

Preprocessing data were conducted on each dataset where the size of the training set, validation set, and testing set were (10,0778, 25,195, and 22,544) respectively in our experiments. The first step was labeling an attack class attribute with a one for a normal and a zero for an attack. In the second step, constant features were removed to save computation costs and avoid an overfitting problem. After that, all traffic samples were scaled and encoded with a Min-Max scalar for numeric features and an ordinal encoder for categorical features in the second step. Lastly, the Analysis of Variance method (ANOVA) was applied as a feature selection to select the most relevant features, which outperformed compared to Spearman's rank, Kendall's rank and Pearson correlation techniques.

5.4. E-BiGAN performances

Although the generative models are used in different fields, there are no agreed methods to measure their performances, especially when used in the cybersecurity field. Yet, cosine similarity and a discrimination score are used to measure the E-BiGAN performance. Cosine similarity is applied to validate the similarity between the generated attacks distribution and the original attacks' distribution based on distance. Therefore, a higher value means that the generated resemble the original attacks.

The discrimination score is another method to measure the E-BiGAN performance, which indicates the ability of the classifier to discriminate between the generated attacks' and original attacks. Thus, Convolutional Neural Networks (CNN) classifier is built to perform this task in this research. The optimal situation for the Discriminator in the GANs model is 0.5 [42]. In other words, the Discriminator cannot distinguish between generated and original attacks. Therefore, lower accuracy in a discrimination score means better attack quality. The architecture used in the discrimination score includes two layers: a 2D convolutional layer and a dense layer. The convolutional has used 40 units with a kernel $size = 3$ followed by a 0.1 dropout and flattened layers. The dense layer has used 40 units and then fed to the last layer for prediction, where the ReLu function has been used in all layers

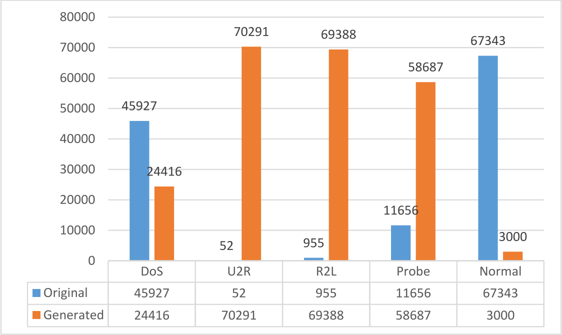

Fig. 3. Data distribution of NSL-KDD.

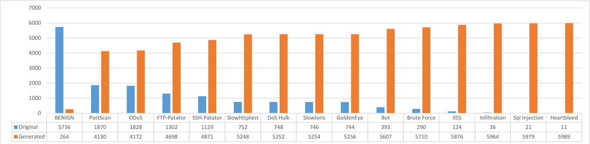

Fig. 4. Data distribution of CICIDS-2017.

except the last layer, which has been the sigmoid activation function. Table 6 shows cosine similarity and discrimination score performance for BiGAN and E-BiGAN using NSL-KDD dataset and CICIDS-2017 datasets.

The results show that the quality of attacks generated by the E-BiGAN model is better than the BiGAN model in both datasets: NSL-KDD and CICIDS-2017 using the cosine similarity metric and discrimination score. The number of original attacks used in all experiments for both datasets is mentioned in Table 2. For generated attacks, the number of attacks used depends on the number of the original attacks per attack category to be balanced. Due to using the best relevant features and dual encoders with sharing weights layer for generators and discriminators, the model has learned better than the BiGAN model, which positively impacts AIDS performance. Figs. 3 and 4 show the total number of original and generated samples used for each attack category in both datasets.

5.5. AIDS architecture

After the E-BiGAN samples are validated and stored, an AIDS is constructed and trained with original and generated samples. The training process depends on the official split, which contains two subsets: training and testing. The CICIDS-2017 dataset has no official split so we split it into training and testing sets, to create a similar split to that in the NSL-KDD dataset.

The training process is applied to the combined samples (original and generated attacks by E-BiGAN), which is now a balanced set of attacks. In AID testing, the model is tested against unseen attacks (testing split), which include attacks that do not exist in the training split. Consequently, the AIDS avoids a bias towards specific attack categories. Fig. 5 shows the process of building an AIDS based on E-BiGAN using NSL-KDD and CICIDS-2017 datasets.

DL algorithms have been used to develop AID based on the E-BiGAN generated data. Therefore, RNN, LSTM, and GRU algorithms have been used in multiple datasets to measure the enhancement of AID performance. RNN algorithm in binary classification includes three layers followed by the prediction layer, which uses the sigmoid activation function. The first layer is SimpleRNN type with 30 input units, while the rest of the layers are dense layers with ten input units and uses a ReLu activation function. The dropout technique has been used with a 0.1 value after the first layer to utilize the model. In multiclass classification, the RNN classifier includes four SimpleRNN layers with ReLu activation functions followed by a dense layer for prediction, which uses softmax. In addition, the dropout technique has been used after each layer with a 0.2 value for each layer.

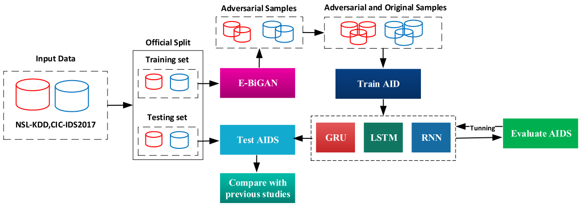

Fig. 5. Process of building AID based on E-BiGAN architecture.

In the GRU algorithm, the architecture used in binary classification includes four layers. The first layer is the GRU layer with 30 input units, while the rest are dense layers with ten input units followed by a 0.1 for dropout. In addition, all layers use the ReLu activation function except the sigmoid activation function's last layer. For multi-class classification, it includes five layers with 30 input units. The first two layers are a GRU layer, while the rest are dense layers. Also, the dropout is used after each layer with a 0.1 value. The same activation functions used in binary are applied in multi-class classification.

For the LSTM algorithm used in binary classification, the architecture includes four layers; the first two layers are an LSTM layer type followed by a 0.5 value for the dropout technique, and the rest are dense layers. The input units for the two layers are 10 and 20, respectively. However, the dense layer includes ten input units for the dense layer. The activation function used is ReLu for all layers except the sigmoid for the last layer. In multiclass classification, the classifier includes four layers; the first three layers are the LSTM layer with ten input units followed by 0.1, 0.2, and 0.1 dropout values after each LSTM layer. The rest layers are dense with ten input units. The same activation function used in binary classification is applied in multiclass classification.

5.6. AID performance metrics

Several standard metrics are used to evaluate an AID's performance. In this research, accuracy, precision, recall, and F1 score are used to evaluate AID based on the E-BiGAN model. Eqs. (11)–(14) shows the formula used for these metrics.

Accuracy (ACC) indicates the percentage of samples that are either normal (negative) or attack (positive) correctly

$$ACC = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{11}$$

Precision shows the percentage of accurately predicted attacks.

$$Precision = \frac{TP}{(TP + FP)} \tag{12}$$

Recall (TPR) shows the percentage of actual attacks.

$$TPR = \frac{TP}{(TP + FN)} \tag{13}$$

F1-Score is the harmonic mean function that fits between the precision and recall values to reflect the performance model without the bias to a precision or recall metric.

$$Fscore = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{14}$$

6. Experiment results

We conducted two main experiments; the first one indicates the impact of E-BiGAN on AID-based models using different DL algorithms (GRU, RNN, LSTM). While the second experiment, the AID based on BiGAN and state-of-the-art models are implemented and compared with the E-BiGAN model to validate the model performance.

Table 7
Statistical analysis of E-BiGAN performance improvements.

Comparison	Metric	Mean difference	p-value	Key finding
vs. BiGAN	Binary Accuracy	+9.0% (98% vs. 89%)	$p < 0.05$	Enhanced overall detection
	Binary F1-Score	+10.0% (99% vs. 89%)	$p < 0.05$	Improved classification balance
	Binary Precision	+9.0% (99% vs. 90%)	$p < 0.05$	Better attack identification
vs. AE	F1-Score	+24.0% (99% vs. 75%)	$p < 0.01$	Significant performance gain
	Precision	+7.0% (99% vs. 92%)	$p < 0.05$	Enhanced precision
	Recall	+35.0% (98% vs. 63%)	$p < 0.01$	Major recall improvement
Attack-Specific	DoS F1-Score	+4.0% (86% vs. 82%)	$p < 0.05$	Consistent DoS detection
	Probe F1-Score	+9.0% (84% vs. 75%)	$p < 0.05$	Enhanced probe detection
	R2L Detection	−6.0% (29% vs. 35%)	$p > 0.05$	Comparable R2L performance
	U2R Detection	+2.0% (24% vs. 22%)	$p > 0.05$	Marginal U2R improvement

Table 8
Binary AID performance compared to state-of-the-art models.

Approach/Metrics	Precision	Recall	Accuracy	F1	ROC (Area)
AE [24]	N/A	N/A	0.83	N/A	N/A
AE [25]	0.92	0.63	0.84	0.75	0.95
AE [29]	0.87	0.98	0.91	0.92	N/A
BiGAN [3]	0.83	0.99	0.89	0.90	N/A
Our BiGAN (GRU)	0.90	0.89	0.89	0.89	0.91
E-BiGAN (GRU)	**0.99**	**0.98**	**0.98**	**0.99**	**0.99**

Table 9
E-BiGAN performance on NSL-KDD for multiclass classification.

Category	Precision		Recall		F1	
	E-BiGAN	AE [25]	E-BiGAN	AE [25]	E-BiGAN	AE [25]
DoS	0.89	**0.97**	0.83	**0.98**	0.85	**0.97**
Probe	**0.84**	0.69	0.83	**0.94**	**0.84**	0.80
R2L	0.82	**0.99**	0.20	**0.39**	0.29	**0.56**
U2R	**0.79**	N/A	**0.14**	N/A	**0.22**	N/A
Normal	0.75	0.85	0.97	0.96	0.85	0.90

6.1. Statistical significance analysis

We conducted a rigorous statistical analysis of our model's performance improvements through multiple experimental runs. Table 7 presents the statistical significance of our results compared to existing approaches.

Key findings from our statistical analysis reveal:

1. **Binary Classification Performance**: The E-BiGAN architecture demonstrates statistically significant improvements in binary classification metrics, achieving 98% accuracy compared to BiGAN's 89% ($p < 0.05$). This improvement is consistent across precision (99% vs. 90%) and F1-score (99% vs. 89%).
2. **Comparison with AE Approaches**: Our model shows substantial improvements over AE-based approaches, particularly in recall (35% improvement, $p < 0.01$) and overall F1-score (24% improvement, $p < 0.01$).
3. **Attack-Specific Detection**: Analysis of individual attack categories reveals:
    - Significant improvements in Probe attack detection (F1-score: 84% vs. 75%, $p < 0.05$)
    - Modest but consistent improvements in DoS detection (F1-score: 86% vs. 82%, $p < 0.05$)
    - Comparable performance in R2L and U2R attacks, with differences not reaching statistical significance ($p > 0.05$)
4. **CICIDS-2017 Performance**: On the CICIDS-2017 dataset, the E-BiGAN model demonstrated notable improvements in detecting various attack types, particularly in DDoS (97% vs. 93%) and DoS Hulk (93% vs. 88%) attacks, while maintaining comparable performance across other categories.

These results demonstrate that the E-BiGAN architecture provides statistically significant improvements in overall intrusion detection performance, particularly in binary classification and common attack detection, while maintaining comparable performance for rare attack classes. The improvements are most pronounced in the model's ability to maintain high precision while substantially improving recall, resulting in better overall F1-scores compared to existing approaches.

6.2. Environment settings

Python 3.6 was used to build the E-BiGAN model and AIDS. It is used due to its availability and stability and is widely used for artificial intelligence applications. A Lenovo laptop (Intel(R) Core(TM) i7-10510U CPU@1.80 GHz) was employed to build E-BiGAN and AIDS. For a Graphics Processing Unit (GPU) device, an AMD Radeon (TM) RX 640 is used to perform processing, training and testing tasks in E-BiGAN and AIDS. All experiments include the same number of samples to evaluate AID based on the E-BiGAN model according to the official split of the NSL-KDD dataset.

6.3. NSL-KDD

The E-BiGAN model is trained on the NSL-KDD with 500 epochs, while the loss values for generators and discriminators are observed to validate the results. A validation set is used to validate that the model has learned, as expected in the training process. Table 8 shows the AID performance using GRU algorithm in binary classification using E-BiGAN architecture compared with our implementation for the BiGAN model and the state-of-the-art models. The BiGAN model is implemented as additional evidence to value the effectiveness of AID based on the E-BiGAN model and generated data.

The results indicate that AID based on E-BiGAN outperforms compared to the state-of-the-art models. Referring to Table 8, our model performance is better than the AID introduced based on AE and statistical tools in binary classification [24,25]. In addition, our model is evaluated and compared to the BiGAN model based on standard metrics. Our model proves the robustness compared to Xu et al. in his research work [29]. Also, our E-BiGAN performs significantly by detecting attacks and delivers high-performance metrics compared with Alptekin et al. [3] model, which trained the encoder and Generator components more than the Discriminator on the BiGAN architecture.

The multiclass classification is conducted to prove that AID based on the E-BiGAN is not biased towards a specific category. Table 9 shows the AID performance model for each category based on E-BiGAN and AE approaches. It can be noticed that the AID-based E-BiGAN architecture is capable of detecting attacks that belong to DoS, U2R, and Probe categories, while the U2R is not considered on AID-AE research by [25]. From an operational perspective, the detection of four attack categories is simpler than the five in AID, where the complexity is increased by the number of classes. Although the results achieved by E-BiGAN are high, the performance for the U2R and Probe categories is much better than AID-AE based on Precision and F1 metrics.

For more investigation, several experiments using DL algorithms (RNN, GRU, and LSTM) have been utilized for AID using BiGAN and E-BiGAN architectures to show the performance among attack categories on the NSL-KDD dataset. The results show that the E-BiGAN effectively detects attacks, especially for Probe, R2L, and U2R, on multiclass classification using RNN, GRU and LSTM algorithms, as indicated in Table 9. It can be noticed that the E-BiGAN outperforms the BiGAN model. The AID based on LSTM and E-BiGAN is the best for all attack categories, while the GRU and RNN are the best for Probe category in terms of F1 score metric. Also, the results show that the RNN is the best algorithm to detect U2R attacks. Table 10 shows the F1 score for AID performance in NSL-KDD using DL algorithms for BiGAN and E-BiGAN architectures.

6.4. CICIDS-2017

This experiment aims to find AID based on E-BiGAN when a dataset includes many attack categories like CICIDS-2017. Also, this experiment aims to explore the E-BiGAN performances compared to the

Table 10
The F1 score for AID based on BiGAN and E-BiGAN on the NSL-KDD dataset.

Category/Algo	BiGAN			E-BiGAN		
	RNN	GRU	LSTM	RNN	GRU	LSTM
DoS	0.88	0.82	0.81	0.85	**0.86**	**0.85**
Probe	0.72	0.74	0.75	**0.81**	**0.83**	**0.84**
R2L	0.36	0.35	0.23	0.25	0.32	**0.29**
U2R	0.20	0.27	0.23	**0.24**	0.23	0.22
Normal	0.85	0.84	0.84	**0.86**	**0.85**	**0.86**

Table 11
F1 score for multiclass classification based on E-BiGAN and BiGAN CICIDS-2017.

Category/Alg	BiGAN			E-BiGAN		
	CNN	GRU	RNN	CNN	GRU	RNN
BENIGN	0.86	0.94	0.88	**0.92**	**0.95**	**0.90**
Bot	0.68	0.76	0.72	**0.75**	0.76	**0.76**
Brute Force	0.00	0.65	0.63	**0.66**	0.65	0.63
DDoS	0.93	0.93	**0.92**	**0.96**	**0.97**	0.90
DoS GoldenEye	0.92	**0.95**	0.91	**0.94**	0.92	**0.93**
DoS Hulk	0.81	0.88	**0.78**	**0.93**	**0.93**	0.76
DoS Slowhttptest	0.77	0.93	0.76	**0.88**	0.93	**0.80**
DoS Slowloris	**0.88**	0.90	0.62	0.69	**0.92**	**0.84**
FTP Patator	0.98	0.98	0.98	**0.99**	**0.99**	0.99
Heartbleed	0.0	0.0	0.0	0.0	0.0	0.0
Infiltration	0.0	0.0	0.0	0.0	**0.25**	**0.14**
PortScan	0.98	0.98	0.97	**0.99**	**0.99**	0.97
SSH Patator	0.98	0.99	0.99	0.98	0.99	0.99
Sql Injection	0.0	0.0	0.0	0.0	0.0	0.0
XSS	0.0	0.0	0.0	0.0	0.0	0.0

BiGAN model for different attack behavior åthis second sentence needs to be better formulated.

The AIDS based on BiGAN and E-BiGAN models were executed in binary classification. The AIDS's performance is almost equal in all DL algorithms, LSTM algorithms achieved the highest performance with a 0.98 % for accuracy, precision, recall, and F1 score. As a result, it can be noticed that the binary classification may not show the robustness of AID for small attack categories since AID may be tricked with unknown attacks.

We conducted a multiclass classification on the CICIDS-2017 dataset using CNN, GRU, and RNN algorithms to show model performance among several learning algorithms, as the dataset includes fourteen attacks. Table 11 shows AID performance based on E-BiGAN and BiGAN models for different DL algorithms åyou are only showing the F1 performance on the detection of different attacks using different DL algorithms.

The results show that the E-BiGAN AIDS outperforms the BiGAN AIDS in terms of F1 metric values in attack detection using CNN, GRU, and RNN algorithms. The best performance is obtained by the CNN algorithm, where the attack detection is enhanced for nine attacks. It is important to note that the E-BiGAN model supports AID in detecting rare attacks such as Bot, Brute Force, and Infiltration. However, the lack of samples for some attacks such as Heartbleed and SQL Injection attacks (fewer than 22 samples) does not help AID, even after applying complex models.

7. Discussion and limitations

Detecting a wide range of attacks is a significant challenge for many reasons. Attack nature, the number of available samples, and the quality of attacks generated by a generative model are common reasons affecting AIDS performance. In this research, all these factors were included in building AID based on E-BiGAN, which delivers high performance compared with previous works.

This research presents a novel method to build a robust AIDS based on the E-BiGAN model supporting binary and multiclass classification. Also, Several DL algorithms are used to prove the capability of the E-BiGAN architecture. The architecture proposed in the E-BiGAN model decreases the dependency between the Generator and Discriminator in generating high-quality (synthetic) attacks. As a result, a wide range of attacks are detected, which can be considered a general and robust solution in the AIDS context.

We tested the E-BiGAN model and compared it with other previous research using standard metrics (Accuracy, Precision, Recall, F1 score) without biased to a specific model on the NSL-KDD and CICIDS-2017 datasets with take consideration of the official dataset split.

Detecting attacks in AIDSs is varied, which shows to which level AIDS can detect attacks with highly accurate results. However, no single solution is capable of detecting all attacks with high performance, especially rare attacks. The rare attacks with limited information lead to low quality in the generative model, which leads to limited performance in AIDS training. In addition, an encrypted connection makes the AIDS limited, which cannot directly inspect the encrypted payload, which is challenging to detect directly. All these limitations leave the research work open for further investigations to enhance AIDS performance.

8. Conclusion and future works

In this research, we introduced the E-BiGAN architecture, which obtains high-quality synthetic attacks than BiGAN and state-of-the-art models. The E-BiGAN model performance is assessed based on the quality of the generated attacks using cosine similarity. Furthermore, the AIDSs constructed based on E-BiGAN attacks are evaluated using two learning approaches (ML and DL), two datasets (NSL-KDD, and CICIDS-2017) and different performance metrics (accuracy, precision, recall, F1, ROC). Over different evaluation metrics, AIDS performance with the E-BIGAN model performs well in both single and multiclass classification. Finally, this research demonstrates that the LSTM algorithm in binary classification and CNN in multiclass classification can deliver a robust AIDS using the E-BiGAN model.

We plan to explore new techniques to enhance limited attack detection performance based on attacks produced by our E-BiGAN model. Such techniques will lead to improved AIDS performance against unseen and rare attacks.

In conclusion, our research demonstrates that the E-BiGAN architecture provides significant improvements in intrusion detection through its novel dual encoder design and weight-sharing mechanism. The experimental results show consistent performance gains across multiple metrics, achieving 98% accuracy and 99% F1-score in binary classification, while maintaining robust performance in multiclass scenarios with up to 89% precision for DoS attacks and 84% for Probe attacks. The architecture's ability to handle rare attack patterns is particularly noteworthy, achieving 79% precision for U2R attacks, a significant improvement over traditional approaches. The model's effectiveness across both NSL-KDD and CICIDS-2017 datasets demonstrates its generalizability to different network environments and attack patterns. Future work could focus on enhancing the model's performance on encrypted traffic and extremely rare attack patterns, potentially through advanced feature extraction techniques or hybrid architectures that combine supervised and unsupervised learning approaches. Additionally, investigating the model's adaptability to real-time traffic analysis and zero-day attack detection presents promising research directions for further improving intrusion detection systems. We plan to explore new techniques to enhance limited attack detection performance based on attacks produced by our E-BiGAN model. Such techniques will lead to improved AIDS performance against unseen and rare attacks.

CRediT authorship contribution statement

Mohammad Arafah: Writing – original draft, Visualization, Validation, Methodology, Formal analysis, Data curation, Conceptualization. **Iain Phillips:** Supervision. **Asma Adnane:** Supervision. **Mohammad Alauthman:** Writing – review & editing, Validation. **Nauman Aslam:** Writing – review & editing.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Lecturer reports a relationship with University of Petra that includes: employment and non-financial support. If there are other authors, they 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] M.A. Ferrag, L. Maglaras, H. Janicke, R. Smith, Deep learning techniques for cyber security intrusion detection: A detailed analysis, in: 6th International Symposium for ICS & SCADA Cyber Security Research 2019 6, 2019, pp. 126–136.

[2] M.S.M. Pozi, M.N. Sulaiman, N. Mustapha, T. Perumal, Improving anomalous rare attack detection rate for intrusion detection system using support vector machine and genetic programming, Neural Process. Lett. 44 (2) (2016) 279–290.

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

[4] R.H. Randhawa, N. Aslam, M. Alauthman, H. Rafiq, F. Comeau, Security hardening of botnet detectors using generative adversarial networks, IEEE Access 9 (2021) 78276–78292.

[5] A. Javaid, Q. Niyaz, W. Sun, M. Alam, A deep learning approach for network intrusion detection system, in: Proceedings of the 9th EAI International Conference on Bio-Inspired Information and Communications Technologies (Formerly BIONETICS), 2016, pp. 21–26.

[6] Y. Peng, G. Fu, Y. Luo, J. Hu, B. Li, Q. Yan, Detecting adversarial examples for network intrusion detection system with GAN, in: 2020 IEEE 11th International Conference on Software Engineering and Service Science, ICSESS, IEEE, 2020, pp. 6–10.

[7] D. Saxena, J. Cao, Generative adversarial networks (GANs): challenges, solutions, and future directions, 2020, arXiv, arXiv:2005.00065.

[8] P. Robic-Butez, T.Y. Win, Detection of phishing websites using generative adversarial network, in: 2019 IEEE International Conference on Big Data (Big Data), IEEE, 2019, pp. 3216–3221.

[9] P. Ge, C.-X. Ren, D.-Q. Dai, J. Feng, S. Yan, Dual adversarial autoencoders for clustering, IEEE Trans. Neural Netw. Learn. Syst. 31 (4) (2019) 1417–1424.

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

[11] X. Zhang, Network Intrusion Detection Using Generative Adversarial Networks (Ph.D. thesis), University of Canterbury, 2020.

[12] V. Kumar, D. Sinha, A.K. Das, S.C. Pandey, R.T. Goswami, An integrated rule based intrusion detection system: analysis on UNSW-NB15 data set and the real time online dataset, Clust. Comput. 23 (2) (2020) 1397–1418, http://dx.doi.org/10.1007/s10586-019-03008-x, http://files/482/Kumar et al. - 2020 - An integrated rule based intrusion detection syste.pdf.

[13] L. Liu, H. Zhang, X. Xu, Z. Zhang, S. Yan, Collocating clothes with generative adversarial networks cosupervised by categories and attributes: a multidiscriminator framework, IEEE Trans. Neural Netw. Learn. Syst. 31 (9) (2020) 3540–3554, http://dx.doi.org/10.1109/TNNLS.2019.2944979, http://files/530/Liu et al. - 2020 - Collocating Clothes With Generative Adversarial Ne.pdf http://files/556/8891673.html.

[14] M. Arafah, I. Phillips, A. Adnane, W. Hadi, M. Alauthman, A.-K. Al-Banna, Anomaly-based network intrusion detection using denoising autoencoder and Wasserstein GAN synthetic attacks, Appl. Soft Comput. 168 (2025) 112455.

[15] S. Sahu, B.M. Mehtre, Network intrusion detection system using J48 Decision Tree, in: 2015 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2015, IEEE, 2015, pp. 2023–2026, http://dx.doi.org/10.1109/ICACCI.2015.7275914.

[16] D.P. Gaikwad, R.C. Thool, Intrusion detection system using Bagging with Partial Decision Tree base classifier, Procedia Comput. Sci. 49 (1) (2015) 92–98, http://dx.doi.org/10.1016/j.procs.2015.04.231.

[17] A.O.. Balogun, R.G. Jimoh, Anomaly intrusion detection using an hybrid of decision tree and K-nearest neighbor, Multidiscip. J. Publ. Fac. Sci. 2 (2015) 67–74.

[18] A. Ahmim, L. Maglaras, M.A. Ferrag, M. Derdour, H. Janicke, A novel hierarchical intrusion detection system based on decision tree and rules-based models, in: 2019 15th International Conference on Distributed Computing in Sensor Systems, DCOSS, IEEE, 2019, pp. 228–233, http://dx.doi.org/10.1109/DCOSS.2019.00059.

[19] S. Waskle, L. Parashar, U. Singh, Intrusion detection system using PCA with random forest approach, in: 2020 International Conference on Electronics and Sustainable Communication Systems, ICESC, IEEE, 2020, pp. 803–808.

[20] M. Asif, S. Abbas, M. Khan, A. Fatima, M.A. Khan, S.-W. Lee, MapReduce based intelligent model for intrusion detection using machine learning technique, J. King Saud Univ.- Comput. Inf. Sci. 34 (10) (2022) 9723–9731.

[21] J.B. Awotunde, F.E. Ayo, R. Panigrahi, A. Garg, A.K. Bhoi, P. Barsocchi, A multi-level random forest model-based intrusion detection using fuzzy inference system for internet of things networks, Int. J. Comput. Intell. Syst. 16 (1) (2023) 31.

[22] J. Li, M.S. Othman, H. Chen, L.M. Yusuf, Optimizing IoT intrusion detection system: feature selection versus feature extraction in machine learning, J. Big Data 11 (1) (2024) 36.

[23] A. Alsajri, A. Steiti, Intrusion detection system based on machine learning algorithms:(SVM and genetic algorithm), Babylon. J. Mach. Learn. 2024 (2024) 15–29.

[24] M. Yousefi-Azar, V. Varadharajan, L. Hamey, U. Tupakula, Autoencoder-based feature learning for cyber security applications, in: 2017 International Joint Conference on Neural Networks, IJCNN, IEEE, 2017, pp. 3854–3861, http://dx.doi.org/10.1109/IJCNN.2017.7966342.

[25] 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, http://dx.doi.org/10.1016/j.neucom.2019.11.016.

[26] S.N. Mighan, M. Kahani, A novel scalable intrusion detection system based on deep learning, Int. J. Inf. Secur. 20 (2021) 387–403.

[27] F. Wiewel, B. Yang, Continual learning for anomaly detection with variational autoencoder, in: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, IEEE, 2019, pp. 3837–3841.

[28] Y. Zhang, Deep Generative Model for Multi-Class Imbalanced Learning (Ph.D. thesis), University of Rhode Island, 2018.

[29] W. Xu, J. Jang-Jaccard, T. Liu, F. Sabrina, J. Kwak, Improved bidirectional GAN-based approach for network intrusion detection using one-class classifier, Computers 11 (6) (2022) 85.

[30] S.M. Kasongo, A deep learning technique for intrusion detection system using a Recurrent Neural Networks based framework, Comput. Commun. 199 (2023) 113–125.

[31] M. Arafah, I. Phillips, A. Adnane, Evaluating the impact of generative adversarial models on the performance of anomaly intrusion detection, IET Netw. (2023).

[32] Chen, et al., Exponential state estimation for delayed competitive neural network via stochastic sampled-data control with Markov jump parameters under actuator failure, J. Artif. Intell. Soft Comput. Res. 14 (2024) 373–3852.

[33] Liu, et al., Input-to-state stability of stochastic Markovian jump genetic regulatory networks, Math. Comput. Simulation 222 (2024) 174–187.

[34] M.S. Aslam, Z. Ma, Output regulation for time–delayed Takagi–Sugeno fuzzy model with networked control system, Hacet. J. Math. Stat. 52 (5) (2023) 1282–1302.

[35] J. Donahue, P. Krähenbühl, T. Darrell, Adversarial feature learning, 2016, arXiv preprint arXiv:1605.09782.

[36] D. Tellez, G. Litjens, J. van der Laak, F. Ciompi, Neural image compression for gigapixel histopathology image analysis, IEEE Trans. Pattern Anal. Mach. Intell. 43 (2) (2019) 567–578.

[37] H. Alqahtani, M. Kavakli-Thorne, G. Kumar, Applications of generative adversarial networks (gans): An updated review, Arch. Comput. Methods Eng. 28 (2) (2021) 525–552.

[38] M. Yousefi-Azar, V. Varadharajan, L. Hamey, U. Tupakula, Autoencoder-based feature learning for cyber security applications, in: 2017 International Joint Conference on Neural Networks, IJCNN, IEEE, 2017, pp. 3854–3861, http://dx.doi.org/10.1109/IJCNN.2017.7966342.

[39] 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, http://dx.doi.org/10.1016/j.neucom.2019.11.016.

[40] W. Xu, J. Jang-Jaccard, T. Liu, F. Sabrina, J. Kabir, Improved bidirectional gan-based approach for network intrusion detection using one-class classifier, Computers 11 (6) (2022) 85.

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

[42] 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).

[43] S. Sapre, P. Ahmadi, K. Islam, A robust comparison of the KDDCup99 and NSL-KDD IoT network intrusion detection datasets through various machine learning algorithms, 1, 2019, http://dx.doi.org/10.13021/jssr2019.2681, arXiv, arXiv:1912.13204.

[44] 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.

[45] S. Choudhary, N. Kesswani, Analysis of KDD-cup'99, NSL-KDD and UNSW-NB15 datasets using deep learning in IoT, Procedia Comput. Sci. 167 (2019) (2020) 1561–1573, http://dx.doi.org/10.1016/j.procs.2020.03.367.

[46] H.K. Aggarwal, M.P. Mani, M. Jacob, MoDL: Model-based deep learning architecture for inverse problems, IEEE Trans. Med. Imaging 38 (2) (2018) 394–405.