International Journal of Computer Applications (0975 - 8887)
Volume 187 - No.15, June 2025

Generative AI-based Intrusion Detection of Imbalanced Network Traffic using Generative Adversarial Variational Auto-Encoder

R. Saranya
Amrita college of Engineering and Technology
Tamil Nadu
India
saranyamvasan@gmail.com

S.L.Jayalakshmi, PhD
Pondicherry University
Puducherry
India

ABSTRACT

Knowledge of system security becomes more and more important as ever-evolving network threats arise. Intrusion detection, a crucial component of cybersecurity, recognizes unusual activity based on traffic patterns. However, harmful cyberattacks can frequently hide enormous amounts of legitimate data in unbalanced network traffic. Generative AI models can be utilized to address this imbalance by generating synthetic data that can improve the development of machine learning models. Traditional intrusion detection systems (IDS) often struggle with imbalanced data, where benign traffic overwhelmingly outnumbers malicious traffic. This imbalance can lead to poor detection rates for rare but significant attacks. To overcome this challenge, a novel approach is proposed using a Generative Adversarial Variational Auto-Encoder (GAVAE) to improve the detection of intrusions in imbalanced network traffic. By combining the probabilistic latent space learning of Variational Auto-Encoders (VAEs) with the adversarial training framework of Generative Adversarial Networks (GANs), the proposed method generates high-quality synthetic samples of minority classes. These synthetic samples augment the training dataset, leading to a more balanced distribution and increased throughput of the intrusion detection model. The proposed model was evaluated on the UNSW-NB15 and NSL-KDD data sets. The experimental results demonstrate that the proposed GAVAE model significantly improves the detection capabilities compared to traditional methods, offering a robust solution for network security.

General Terms

Intrusion Detection, Variational Auto-Encoder

Keywords

Intrusion Detection, Auto-encoder, Imbalanced network traffic, Generative adversarial Network (GAN), Deep Neural Network

1. INTRODUCTION

Network security is becoming more complex due to the emergence of modern Internet technologies such as file sharing, mobile payment systems, and the growing presence of devices on the Internet of Things (IoT) [1, 2, 3]. Specifically, the requirement for network security is increasing because of the necessity to safeguard private data, a rise in hacking occurrences, a rise in the conversion of personal computers into zombie PCs in response to the growth in open information system users, the quick information sharing among hackers, and the sharp rise in Internet usage. As a result, a network intrusion detection system (NIDS) is now a key part of network security and computing.

An irregularity in a network is the sign of an intrusion or threat. Hackers utilize bugs in software such as buffer overflows and poor security standards among other network vulnerabilities to their advantage, weakening the security of the network. Hackers, often ordinary Internet users, attempt to steal or compromise sensitive data from a victim's system. These intruders may include external attackers or authorized users with limited privileges seeking to gain higher access rights. Intrusion detection techniques are generally classified into two main types: anomaly detection and signature detection. By comparing the network's packet flow with the previously established, configured known signatures of known attacks, signature-based detection keeps an eye on known threats. On the other hand, attacks are identified using the anomaly detection technique, which compares events that indicate a departure from the authorized user parameters with those that have been set [2]. When malicious activity occurs on a network, the intrusion detection system (IDS) creates logs and notifies the network administrator [2, 3, 4].

One important safeguard for protecting computers and network infrastructures against misuse is intrusion detection [3]. Intrusion detection systems (IDS) typically operate using two main techniques: anomaly detection, which monitors and flags activities that deviate from established normal patterns, and signature-based detection, which identifies threats by matching them to a database of known attack signatures. To improve detection accuracy—especially for unknown threats—and to minimize false positives when recognizing familiar attack patterns, recent research has focused on hybrid detection methods. These combine both anomaly-based and signature-based approaches to enhance overall system security [4].

Despite their widespread acceptance, about 80 % of IoT devices are susceptible to various cyberattacks [4]. They are susceptible to various attacks, such as denial-of-service (DDoS), unauthorized device access, data breaches, identity theft, and man-in-the-middle exploits. Robust security methods that can identify both known and new attacks must be considered in order to protect critical systems critical to security against attackers [4, 5, 6]. Primarily, intrusion detection systems (IDSs) are the first line of defense in the CPS domain. They are in charge of monitoring system data and network traffic for malicious activity and sending out alarms.

Subsequently, researchers applied machine learning methods to identify intrusions [1]. At that time, however, these techniques were not widely adopted due to constraints in computational power and data storage capabilities. The rapid progress in computing and the growing influence of artificial intelligence (AI) have led researchers to incorporate machine learning techniques into network security to improve threat detection and system protection. They have obtained certain outcomes [2, 3, 4].

To identify anomalies in the network, an IDS integrated with machine learning model will support to achieve an improved accuracy [7]. While IoT, Big Data, Cloud computing, and Industry 4.0, are some of the rising IT developments in CPS that are gaining popularity, they are also creating new risks [8]. In addition, new architectural arrangements are making the model more complex because of unidentified emergent behavior [9]. It is necessary to deploy each IDS individually to examine how they interact with this complicated system; yet, the model training is being hampered by a lack of data. Furthermore, the majority of these publicly accessible datasets are imbalanced, meaning that certain categories of attack data are scarce relative to normal data.

To overcome the limitations of current systems, the proposed approach introduces an intrusion detection system (IDS) enhanced by a generative adversarial network (GAN), aiming to reduce dataset-related constraints associated with emerging technologies. Due to their lack of extensive training datasets and technological limitations for processing large amounts of data, traditional artificial neural networks (ANNs), which were previously used to do this task, are no longer useful. Due to the unique features of the present internet, it has been decided to use the capabilities of contemporary artificial neural networks (ANN) to detect security intrusions. In this paper, the suitability of generative adversarial networks (GANs) is examined for detecting security breaches in large-scale cyber device networks.

In this work, the popular techniques' limited ability to self-learn in the current intrusion detection systems and the absence of comprehensive and reliable datasets mean that they are unable to fully solve all intrusion detection tasks for m2m networks. A relatively recent class of artificial neural networks called Generative Adversarial Networks (GAN) is primarily focused on producing specific data [7]. The GAN is made up of two neural networks that work together: the generator (G), which creates artificial data, and the discriminator (D), which evaluates these data by comparing them to real samples. These networks engage in a competitive process—G strives to generate data that closely resembles real inputs, while D enhances its ability to detect fake data. The information flow from G to D is known as the channel, and the feedback path from D back to G is called the return path. The main objective of GAN-based models is to gain a complete understanding of the hidden patterns within the data. This makes them especially useful for addressing challenges related to insufficient or incomplete datasets.

Compared to conventional ANNs, GANs offer several significant advantages. They can efficiently detect and classify unusual network behaviors and generate additional synthetic anomalies to improve the quality of flagged data. GANs also produce samples more rapidly than fully visible belief networks—such as Neural Autoregressive Distribution Estimation (NADE), Pixel Recurrent Neural Networks, or WaveNet—because they do not require step-by-step generation of each sample component. Unlike Boltzmann machines, which depend on Monte Carlo methods to approximate gradients of the log partition function, GANs can be trained without such complex estimations, making them easier to implement. Furthermore, unlike variational autoencoders, GANs avoid introducing deterministic bias during training.

In real-world network environments, the vast majority of traffic originates from legitimate user activity, while malicious attacks represent only a small portion. This leads to a significant imbalance in data categories, where normal traffic overwhelmingly dominates. Such an uneven distribution creates challenges for intrusion detection systems, as cyberattacks can easily be masked within the large volume of routine traffic. This makes accurate classification difficult, often resulting in misidentification, and hampers the ability of machine learning models to learn the characteristics of the minority (attack) classes effectively [5]. Leveraging adaptive learning to distinguish between typical and abnormal behavior using large datasets can enhance real-time intrusion detection capabilities. However, despite these efforts, classification imbalances continue to hinder the accurate multi-class detection of network threats.

In this work, a unique generative adversarial variational autoencoder (GAVAE) approach is proposed to address the issue of class imbalance in network traffic when faced with unbalanced data. This method effectively addresses data imbalance and strengthens the classification model's ability to learn difficult examples. To assess the performance of the proposed approach, experiments were conducted on two widely used benchmark datasets, utilizing both deep learning techniques and conventional machine learning methods. The key contributions of this work are outlined below.

—To perform a thorough analysis and data cleaning on two benchmark datasets: the traditional NSL- KDD and UNSW-NB15.

—To handle the problem of class imbalance in intrusion detection and enhance the model's capability to distinguish between different classes during training, this study introduces an innovative approach called GAVAE. This method reduces the number of majority class samples while increasing the minority class instances within the difficult subset of the data.

The rest of the article is presented as below. The relevant work in the area of intrusion detection is analyzed in Section II. Section III suggests a modified model and provides necessary background data. The intrusion detection system suggested is described thoroughly in Section IV. Section V displays the results and discussions for the suggested model in comparison to the most advanced classification models and established approaches. In Section VI, this article concludes with the conclusion.

2. RELATED WORK

Intrusion Detection Systems (IDS) are classified into two main types: Host-Based and Network-Based. Host-based Intrusion Detection System (HIDS) is used by administrators to monitor and analyze the behavior and events occurring on a specific host or device. A key benefit of HIDS is its capability to analyze encrypted data as it moves across a network. However, managing HIDS can be challenging since each host requires individual configuration and oversight. Moreover, some denial-of-service attacks may disable HIDS. In contrast, Network-based Intrusion Detection System (NIDS) is a hardware- or software-based system strategically positioned within a network to monitor traffic without directly interacting with the devices it oversees.

Network-Based Intrusion Detection Systems (NIDS) function with two interfaces: one for management and reporting, and another for network traffic monitoring. One of the main benefits of NIDS is its capability to oversee extensive networks with minimal hardware deployment. Moreover, since NIDS typically operate discreetly, they provide an added layer of security by remaining hidden from potential attackers. However, a significant drawback is their struggle to precisely identify attack vectors when network traffic is exceptionally high.

The data-level based network intrusion detection model is examined in [2]. This study systematically develops three data-driven research approaches: a data augmentation method utilizing a variational autoencoder (VAE), a balancing strategy employing a conditional VAE, and a combined technique that integrates conditional VAE with random under-sampling to address data imbalance. The deep-learning-based IDS is integrated with the three data-level-based schemes. An unsupervised deep learning approach combined with a semi-supervised learning strategy is utilized to identify anomalous network traffic or intrusions from flow-based data is used in [3]. More precisely, flow features were used to identify unknown attacks using Autoencoder and Variational Autoencoder algorithms. The experimental findings demonstrate that Variational Autoencoder outperforms Autoencoder and One-Class Support Vector Machine in most cases.

To identify intrusions within imbalanced network traffic, [4] explores the use of both deep learning and traditional machine learning methods. A new technique called the Difficult Set Sampling Technique (DSSTE) is introduced to address the issue of class imbalance. Furthermore, [5] describes an intrusion detection model using CNN architecture. Prior to training, the network traffic data is balanced using the SMOTE-ENN (Synthetic Minority Oversampling Technique combined with Edited Nearest Neighbors) method. The NSL-KDD dataset is used to analyze the performance of the proposed model.

Three categorization strategies were employed in the work of Alkasassbeh and Almseidin [6] to address the low accuracy problems that are frequently encountered by IDS that use artificial neural networks with fuzzy clustering for handling infrequent attacks. By dividing the heterogeneous training data set into homogeneous subsets, they were able to successfully increase accuracy while lowering the complexity of each training set. The suggested work used J48 trees, Multilayer Perceptron (MLP), and Bayes network techniques, with J48 trees providing the best accuracy. Their inability to use feature selection to eliminate all unnecessary, redundant, and undesirable features is a significant flaw in their work.

By utilizing a voting classifier to combine the output of several supervised and unsupervised machine learning methods, Marilyn Z. and Chung-Horng L. [7] developed an ensemble-based approach to IDS. The study improves the performance and accuracy of the available intrusion detection systems. This research enhances the effectiveness and precision of existing intrusion detection systems. The authors opted for the Kyoto2006+ dataset, which, while older, is considered more reliable than the widely used KDDCup '99 dataset. As a result, their approach aims to achieve a certain level of accuracy. However, in some cases, the recall rate is relatively low, suggesting an elevated false negative rate (FPR).

To overcome the limitations of individual classifiers, [8] introduces an ensemble-based method. This approach features a scalable and efficient ensemble model that relies on majority voting, enabling real-time analysis of network traffic and early detection of potential threats. An efficient model was created by taking into account the characteristics of current machine learning techniques. In [8], the proposed intrusion detection methodology is based on ensembles. The suggested model uses decision trees, logistic regression, and naive Bayes as voting classifiers. The effectiveness of the model was assessed using a number of well-known, cutting-edge approaches currently in use. Additionally, an analysis of the suggested model's efficacy was conducted using the CICIDS2017 dataset. The outcomes show a notable increase in accuracy.

A technique to assess the danger of adversarial assaults on ML-based IDS that makes use of generative adversarial networks and active learning is proposed in [11]. This approach gets around these drawbacks by showing how to compromise an IDS with sparse training data and presuming that the only thing known about the IDS model beforehand is its binary classification. Conventional machine learning methods often face challenges such as data imbalance and redundant features when processing complex and large-scale network data. These issues contribute to poor detection accuracy, increased false alarm rates, and inadequate real-time performance in intrusion detection systems. To tackle these problems, [12] proposes a Convolutional Neural Network-based Intrusion Detection Method focused on addressing data imbalance, referred to as CNNIDM-DI. Conventional statistical learning techniques, including Naive Bayes [7], Decision Trees [8, 9, 10], Random Forests [9, 10, 11, 12, 13, 14], and Support Vector Machines [10, 11, 12, 13, 14], are used in the majority of previous works to develop intrusion detection. Numerous research efforts have applied neural network models to intrusion detection, including Multilayer Perceptrons [15], Convolutional Neural Networks (CNNs) [16], and Recurrent Neural Networks (RNNs) [17], inspired by the significant advancements achieved through deep learning. Moreover, Aljawarneh et al. [18] introduced a hybrid approach that incorporates feature selection, resulting in improved accuracy for intrusion detection systems.

The efficacy of most intrusion detection systems is hampered by class-imbalanced data, despite the significant advancements gained in previous approaches [19]. Class imbalance poses a major challenge in intrusion detection when the number of attack instances is significantly lower than that of normal traffic. Deep learning approaches, which often deliver results on par with or better than human performance, have seen widespread adoption in fields like computer vision, natural language processing, speech recognition, pharmaceutical research, and cybersecurity. Popular deep learning architectures employed in these areas include CNN, Long Short-Term Memory (LSTM) networks, and Generative Adversarial Networks (GANs) [8, 9, 10, 11]. These technologies still have a lot of issues, though. First, there is an imbalance in the sorts of attack traffic due to the quick advancement of network technology and traffic data.

Conventional classification methods struggle to achieve high detection accuracy when working with imbalanced datasets. Additionally, they relied on advanced technologies such as AI which has led to an increase in unidentified attacks, posing a greater threat to the security of the networks. Although conventional classifiers efficiently detect known threats, their ability to recognize previously unseen attacks remains limited. Furthermore, the rapid expansion of IoT devices and the widespread use of cloud computing have significantly increased both the volume and complexity of network traffic, making it more difficult for traditional methods to differentiate between legitimate and malicious behavior.

In summary, although existing deep learning models have demonstrated promising performance in network intrusion detection, they often fail to accurately detect rare or unknown attacks. To overcome these limitations, a new hybrid intrusion detection framework is proposed. To generate novel attack samples, this approach involves combining encoded latent vectors with corresponding attack data and sending them to a decoder. The model leverages GAVAE to learn the latent features of intrusion data, helping to balance the data set and improve the variety and quality of the training samples. In order to create a DNN classifier, the GAVAE encoder also includes a softmax layer. Lastly, the DNN classifier efficiently identifies unknown threats from the network attack data by automatically examining high- level abstract feature representation.

3. BACKGROUND

The ideas of autoencoders and GAN, which are essential parts of the anomaly detection method, are briefly explained in this section.

3.1 Auto-Encoder

One of the core deep learning models in artificial neural network called the autoencoder [20, 21], is trained by an unsupervised learning procedure. Restoring the output to the original input as closely as feasible is the aim of autoencoders. Therefore, in order to reduce the reconstruction error, the parameters are gradually updated throughout the training phase. An autoencoder consists of two fundamental components: an encoder and a decoder, as illustrated in Fig 1. The encoder is primarily responsible for reducing the dimensionality of the input data by transforming it into a latent representation. In contrast, the decoder functions as a reconstruction mechanism, capable of generating new data or removing noise from raw input. The process of converting input data (x) into its corresponding latent space representation using the encoder is described in Eq.1.

$$z = f(xW + b) \tag{1}$$

where W represents the weight matrix and b denotes the bias vector, while f refers to the activation function used by the encoder. On the other hand, the decoder's job is to as closely as possible rebuild the representation z into the matching input data (Eq.2)(i.e., $\tilde{x}$).

$$\tilde{x} = g(zW' + b') \tag{2}$$

Here, g denotes the activation function applied by the decoder, while and b' represent the decoder's weight matrix and bias vector, respectively. As a result, training the autoencoder to reduce reconstruction error L_{RE} as shown in Eq.3.

$$(x, \tilde{x}; W, W') = \|x - \tilde{x}\|^2 = \|x - g\left(W'(xW + b) + b'\right)\|^2 \tag{3}$$

One of the primary functions of an autoencoder is to transform high-dimensional input data into a lower-dimensional form that retains the most essential and relevant features. Representing high-dimensional input data as lower-dimensional information (summarized yet relevant information) is one of the autoencoder's core functions. Here, autoencoders are used to extract features from the input data (i.e., reduce its dimension). Although high- dimensional data is typically projected into a lower dimensional space using PCA, the autoencoders are used to perform nonlinear transformations on complicated data sets.

3.2 Generative Adversarial Networks

Generative models aim to replicate the probability distribution of a training dataset to create synthetic data that closely resembles real data. Among these models, Generative Adversarial Networks

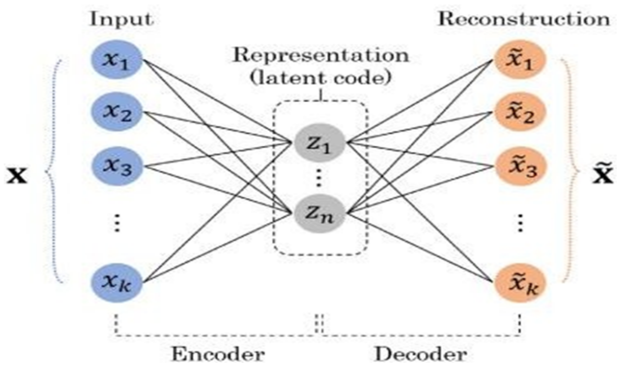

Fig. 1. Auto-Encoder architecture

(GANs) [4] have recently gained significant interest. As a result, numerous variations of GANs have been developed to improve both performance and functionality [22, 23]. A typical GAN, illustrated in 2 consists of two neural networks: a generator (G) and a discriminator (D). The generator's role is to produce realistic synthetic data, while the discriminator attempts to differentiate between genuine and generated samples. In essence, the training process involves a competitive dynamic between these two networks, each working to outperform the other.
In order to produce synthetic data that is nearly identical to the real data (training data), generative models are intended to mimic the probability distribution of a training data set. Among these generative models, GAN research [4] has attracted a lot of attention lately. As a result, several GAN models have been put out in an effort to enhance functionality and performance [22, 23, 4]. Two models based on neural networks make up a GAN model as shown in Fig. 2 a generator G and a discriminator D. While the discriminator D seeks to distinguish between real and fake data, the generator G seeks to produce synthetic data (fake data) that is similar to the real data. Put another way, during the training process, these two elements have competing goals.

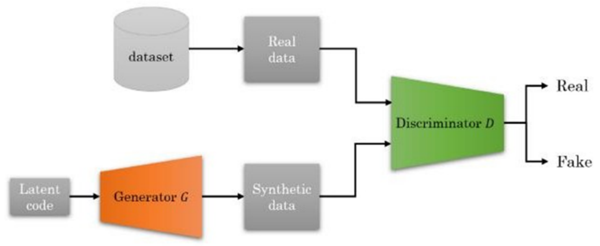

Fig. 2. Architecture of Generative adversarial Network

Formally speaking, it has been defined p_z and p_{data} as the probability distributions of the real data and the latent code, respectively. Then, a GAN's objective function V(D, G), which is made up of a discriminator D and a generator G, is a minimax game and may be expressed in the Eq. 4 as follows:

$$\begin{aligned} V(D,G) = \min_G \max_D \; & E_{\mathbf{x}\sim p_{\mathrm{data}}}\left[\log D_{\theta_D}(\mathbf{x})\right] \\ + \; & E_{\mathbf{z}\sim p_z}\left[\log\left(1 - D_{\theta_D}\left(G_{\theta_G}(\mathbf{z})\right)\right)\right] \end{aligned} \tag{4}$$

where the model parameters of D and G are indicated by $\theta_{\mathbf{D}}$ and $\theta_{\mathbf{G}}$ respectively. As a result, the generator learns to produce synthetic data that can increase the likelihood of being classified as real by the discriminator, while the discriminator is trained to assign higher confidence scores to genuine data. Through continuous iterations of this adversarial training process, both networks improve their performance until they eventually reach a state of equilibrium, where neither can further enhance its results.

3.3 Generative Adversarial Auto- Encoder

Fig. 3 illustrates the structure of the Generative Adversarial Auto-Encoder (GAAE). Since the Adversarial Auto-Encoder (AAE) shares functional similarities with the Variational Auto-Encoder (VAE), the latent variable z in the Auto-Encoder is constrained to follow a specified distribution method. In the case of AAE, z can be declared and sampled as input for the discriminator. Unlike VAE, which optimizes the evidence lower bound (ELBO), AAE employs a Generative Adversarial Network (GAN) to align the encoder's output distribution $q(z \mid x)$ with the target prior p(z). The discriminator is modeled to discriminate between samples drawn from the true prior p(z) and the encoder-generated latent representations z.

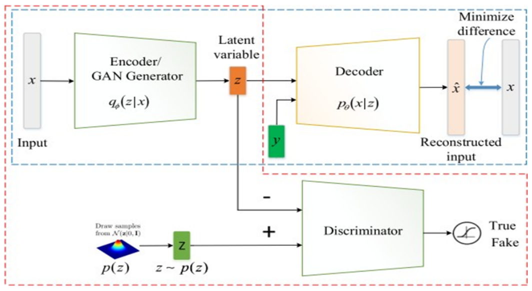

Fig. 3. Adversarial Auto-Encoder Framework

The regularized adversarial auto-encoder incorporates label information directly into the adversarial training process. To give the latent variables a better fit distribution, the discriminator network receives the one-hot label. As shown in Fig.3, incorporating a one-hot encoded label into the decoder's input effectively transforms the training process from unsupervised to supervised. In the GAAE framework, the hidden variable z is combined with the one-hot label y and used together as input to the decoder. Next the network is tuned to acquire the output label-independent latent distribution. While this is not achievable in AAE, new attack samples can be created with the designated labels in GAAE.

3.4 Proposed GAVAE Intrusion Detection Framework

The proposed Generative Adversarial Variational Autoencoder (GAVAE) is illustrated in Fig. 4, which combines the benefits of VAE and GAAE. The primary distinction between the GAAE model and its predecessor is that the former employs AE to characterize the latent distribution of attack samples, whilst the latter uses VAE.
Furthermore, both the discriminator and the decoder are adjusted by using one-hot encoded class vectors. Its primary objective is to control how independent the latent variables are from the class labels. The conditional distribution ($q(x \mid z)$, y) allows the decoder to generate attack samples corresponding to a specific label y. The

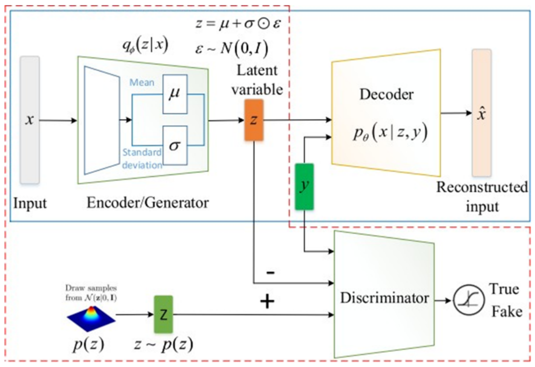

Fig. 4. Proposed GAVAE architecture

proposed system consists of four stages: Preprocessing, Training GAVAE , Data Augmentation, and Intrusion detection.

3.5 Preprocessing

The system uses the preprocessing module, which consists of three subprocesses, to purify a given raw data source before creating and training AI models. 1) Feature scaling; 2) One-hot encoding; and 3) outlier analysis. In the outlier analysis phase, the system filters out abnormal data points that might negatively impact the model's training performance. Outliers are typically found by using reliable metrics of size to quantify the statistical distribution of the data sets. To identify outliers, several commonly used robust measures of scale are available, including the median absolute deviation (MAD) and the interquartile range (IQR). Among these, MAD was selected. The MAD of a numerical property A = x1, x2,..., xn is defined (Eq.5) as follows:

$$\mathrm{MAD} = \mathrm{median}\left(|x_i - \mathrm{median}(A)|\right)$$

It is assumed that the data set's numerical properties have a normal distribution. Consequently, 1.4826 × MAD is a consistent estimator *'q'* for estimating the standard deviation. Using this estimator, it has been concluded that values more than ($10 \times \hat{\sigma}$ are considered outliers for a given numerical attribute. Naturally, outlier analysis is carried out separately for each class and solely on the numerical attributes. Keep in mind that eliminating outliers should come before scaling features because the latter can hide outlier-related information. The system converts nominal qualities into one-hot vectors after removing the outliers. Each categorical attribute is encoded using a binary vector whose length matches the number of possible attribute values. The position representing the actual value is set to 1, while all other positions are assigned a 0. For instance, the attribute "protocol," which is frequently present in network traffic data, is converted into a binary vector of length three when the values TCP, UDP, and ICMP are present. The resulting values are then represented as [1, 0, 0], [0, 1, 0], and [0, 0, 1], respectively. The system scales the numerical properties in conjunction with the one-hot encoding procedure. Scaling for numerical features can often be attributed to normalization and standardization [24]. Between the two strategies, the min-max normalization strategy 2 was selected. For a numeric attribute A, the normalization function $f_A(\cdot)$ that maps $\forall x \in A$ into the range $[0, 1]$ is defined in Eq.6:

$$f(x_i) = \tilde{x}_i = \frac{x_i - \min x_j}{\max x_j - \min x_j} \quad \text{for } x_i \in A \tag{6}$$

where the symbol x_i stands for the i^{th} value (or element) of set A value. Generally speaking, feature extraction (PCA, Pearson correlation coefficient, etc.) is taken into consideration at this stage in deep learning-based approaches in order to feed the model with as many informative features as possible. As a result, feature extraction can have a major impact on how well models perform in anomaly detection. Computational feature extraction is excluded from consideration, since the framework integrates an autoencoder model capable of learning representations without the need for manual feature extraction. It should be noted that, in the proposed framework, there was no noticeable difference between the model with and without a computational feature extraction approach. Later on, a thorough explanation of how to use the autoencoder as a feature extractor is provided.

4. TRAINING GAVAE

The training procedure for the proposed GAVAE involves the following steps for each minibatch: First, the Variational Autoencoder (VAE) is optimized by minimizing the binary cross-entropy loss, aiming to reduce the discrepancy of the recreated output $\hat{x}$ and the original input x.

—(1) The discriminator is trained to distinguish between latent representations drawn from the actual data distribution a(z) and those sampled from the prior Gaussian distribution b(z), using the WGAN-GP loss as the objective function to guide this differentiation.

—(2) The encoder, acting as the generator, is trained to fool the discriminator by producing latent representations z that are nearly indistinguishable from those sampled from the true data distribution.

After training SAVAER, the actual data (x,y) is passed through the trained VAE to compute the reconstruction loss using binary cross-entropy. For each sample (x_i, y_i) the binary cross-entropy loss is estimated as described in Eq.7.

$$L(x_i, y_i) = -\sum_{k=1}^{d} [x_{i,k} \log \hat{x}_{i,k} + (1 - x_{i,k}) \log(1 - \hat{x}_{i,k})] \tag{7}$$

where k represents the k-th feature of the each input sample; d represents the number of features. The reconstruction loss with maximum value for class maxL_j of class j is defined in the Eq.8 as follows:

$$\max L_j = k \cdot \max\{L(x_i, y_i) \mid y_i \in \text{class } j\} \tag{8}$$

where K is the scaling factor and it is set to 1.0.

4.1 Data Augmentation

To enhance the classifier's ability to detect unknown and minority attacks, new attack samples can be generated during the data augmentation phase using a trained decoder as shown in Fig. 5. This process involves sampling the latent variable z_{new} from a multivariate Gaussian distribution p(z), combining it with the corresponding minority class label y_{new}, and feeding the resulting input into the decoder to create new attack instance x_{new}.

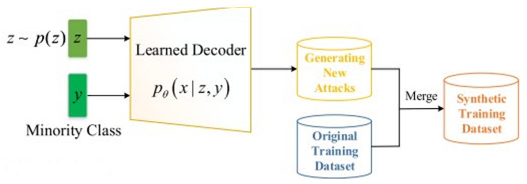

Fig. 5. Data Augmentation

The reconstruction error L(x_{new}, y_{new}) of the newly generated sample is determined using Equation 8 once the newly generated sample (x_{new}, y_{new}) is fed into the trained GAVAE. Filter the newly created attack samples (x_{new}, y_{new}) in accordance with Eq.9 to exclude the samples that change noticeably from the original sample distribution. This approach guarantees that the newly synthesized samples follow the same spatial property as the original data. To achieve a balanced training dataset, these newly generated attack samples—aligned with the original data distribution—are ultimately integrated into the original training set.

$$S = \begin{cases} S \cup \{x_{\text{new}}, y_{\text{new}}\}, & \text{if } y_{\text{new}} \in \text{class } j \\ \& \; L(x_{\text{new}}, y_{\text{new}}) \leq \max L_j, Otherwise & \end{cases} \tag{9}$$

4.2 Intrusion Detection

In the attack detection stage, the trained GAVAE encoder is extended with a softmax layer on top of its final layer to form a deep neural network classifier as shown in Fig 6. The weights of the DNN's hidden layers are initialized using the weights of the learned GAVAE encoder. The DNN classifier is trained using the synthetic training dataset. The DNN classifier's hidden layer weights are first frozen, followed by backpropagation to update the output layer's weight, the unfreezing of all hidden layers, and the use of the augmented training data set to refine the classifier.

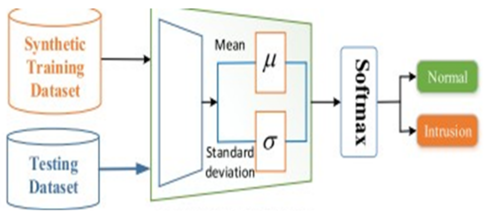

Fig. 6. Detecting attacks

5. EXPERIMENT AND DISCUSSION

All experiments are carried out on a ThinkStation workstation featuring an Intel E5-2620 processor and 64 GB of RAM, operating on a 64-bit Windows 10 system within a TensorFlow framework. To assess the effectiveness of the proposed model alongside several commonly used classifiers, three benchmark datasets are employed: UNSW-NB15 [25], NSL-KDD (KDDTest+) [24], and NSL-KDD (KDDTest-21) [24]. The model's architecture and hyperparameters are fine-tuned through a grid search strategy combined with 5-fold cross-validation to achieve optimal prediction performance.

5.1 Dataset description

To assess the effectiveness of an intrusion detection approach, it is essential to utilize appropriate datasets. In this study, the NSL-KDD dataset [24] and the UNSW-NB15 dataset [25] are employed to evaluate the performance of the GAVAE-IDS model.

5.1.1 NSL-KDD dataset. NSL-KDD [24] is a vast collection of network records that serves as a standard for intrusion detection techniques. These are popular datasets for assessing IDS; UNSW-NB15 has 2,540,044 samples, while NSL-KDD has 148,517 samples. Table 1 lists the 41 features that are present in each NSLKDD sample. These characteristics fall into four groups: host-based features, time-related features, content-related features, and basic features. Additionally, NSLKDD was pre-segregated into two sets: KDDTrain+ and KDDTest+, which correspond to the training and test sets.

Table 1. Description of Traffic Feature Groups

Attributes	Description
1–9	Fundamental features of network connections
10–22	Content-related features
23–31	Time-related features
32–41	Host-based features

The NSL-KDD dataset consists of both normal records and attacks. The assaults are divided into four groups: probing, user 2 root (U 2 R), remote 2 user (R 2 L), and denial of service (DOS). The five classes in the NSL-KDD dataset are DOS, R 2 L, U 2 R, Probe, and Normal. Below are the specifics of the attack categories [25]:

(1) Denial of Service (DoS): This category includes Distributed Denial-of-Service (DDoS) attacks like Smurf, Ping of Death, and Neptune. Such attacks are designed to overwhelm system or network resources, preventing them from responding to legitimate requests or causing them to become unresponsive.
(2) Remote 2 User (R2L): R2L attacks occur when an attacker gains unauthorized access and uses that access to send packets across a network. The attacker can quickly use programs like Guess_ password, Xclock, and others to obtain local access as the machine's user.
(3) User 2 Root (U2R): This type of attack happens when an attacker tries to gain access to a regular user's account or system, such as Xtrem, Perl, etc.
(4) Probing: An attack known as probing occurs when an unauthorized person screens a system to learn more about the network. This information may also be utilized in future attacks or to go over security measures like Satan, mscan, and so forth.

5.2 UNSW-NB15 DATASET

The UNSW-NB15 dataset, introduced by Moustafa et al. [19] in 2015, comprises a diverse range of attack types, making it well-suited for evaluating the performance of various intrusion detection systems. Each entry in the dataset includes 42 features, categorized into basic features, content-based attributes, time-related properties, general-purpose characteristics, and connection-based metrics, as outlined in Table 2. However, both the UNSW-NB15 and NSL-KDD datasets suffer from class imbalance. As shown in Table 3, the imbalance ratios in NSL-KDD vary from 1.44 to 305.77, whereas UNSW-NB15 exhibits even greater disparities, with ratios ranging from 10.30 to 12,751.50. Additionally, the NSL-KDD test set contains 38 types of anomalies, compared to only 23 types in the training set, further highlighting the challenge of imbalance in these datasets.

Table 2. The different categories of features in the UNSW-NB15 dataset

Attributes	Description
1–13	fundamental characteristics of network connections
14–21	content-based aspects of the traffic
22–30	Time-related features
31–35	General purpose traffic metrics
36–42	connection-based features

Rather than using binary classification, the NSL-KDD dataset is evaluated with a 5-class classification approach, while the UNSW-NB15 dataset is assessed using a 10-class classification setup. The intrusion detection problem is framed as a multiclass classification task by employing multiple One-vs-All strategies. In each strategy, a single class is designated as the positive class, and all other classes are grouped as negative, enabling focused detection for each individual category.

Table 3. Class Distribution and Imbalance Ratios in NSL-KDD and UNSW-NB15 Datasets

Dataset	Class	Samples	Imbalance ratio
NSL-KDD	Normal	77,054	–
	DoS	53,385	1.443
	Probe	14,077	5.474
	R2L	3,749	20.553
	U2R	252	305.770
UNSW-NB15	Normal	2,218,761	–
	Generic	215,481	10.297
	Exploits	44,525	49.832
	Fuzzers	24,246	91.510
	DoS	16,353	135.679
	Reconnaissance	13,987	158.630
	Analysis	2,677	828.824
	Backdoor	2,329	952.667
	Shellcode	1,511	1,468.406
	Worms	174	12,751.500

To mitigate the issue of class imbalance in the NSL-KDD dataset, the GAVAE module is utilized to generate additional samples specifically for the four underrepresented classes: DoS, Probe, R2L, and U2R. Fig. 7 displays the D convergence curves based on each class's accuracy. The curves show that, after 4000, 5000, 6000, and 6000 iterations, respectively, the module has been well-optimized to produce samples for DoS, Probe, R2L, and U2R. The UNSW-NB15 and NSL-KDD optimization processes follow the same steps and provide convergence curves that are comparable.

Taking the DoS curve in Fig. 7(a) as an example, during the initial 1000 training iterations, the discriminator (D) could easily differentiate between real and generated samples because the generator (G) had not yet learned to produce realistic representations. As a result, D's initial Accuracy score was nearly 100 %. Between the 1000th and 2000th iterations, D's accuracy decreased from 100 % to 20 %, primarily due to G's rising generative capacity, which made it more difficult for D to distinguish between synthetic and genuine samples. D and G continued to optimize one another for the final

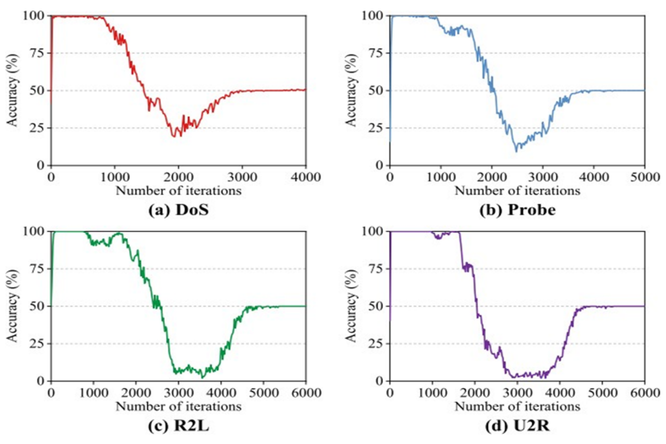

Fig. 7. Convergence curve of discriminator D

2000 iterations until they achieved an equilibrium, at which point D lost its ability to discriminate and converged to 0.5.

The Probe curve is shown in Fig. 7(b), where Accuracy almost approached 100 % in the first 1000 iterations due to G's poor sample generation ability. Because G's generating power continued to increase, the Accuracy roughly decreased from 100 % to 10 % in the 1000th to 2500th iterations. D and G continued to optimize one another during the final 2500 iterations until D reached 0.5. Regarding R2L and U2R (Fig. 7(c) and (d)), G's accuracy, which reached about 100 % in the first 2000 iterations, was insufficient to produce samples. As G's generative capacity continued to rise, the Accuracy roughly decreased from 100 % to 5 % in the 2000th to 4000th iterations. D and G continued to optimize one another during the final 2000 iterations, until D eventually converged to 0.5.

As mentioned, the DNN module uses both real and generated samples as input for training. The DNN module performs the last intrusion detection on new data after it has been properly tuned. The Deep Neural Network (DNN) is optimized using the Adam optimizer with a learning rate set to 0.00005, while Categorical Cross-Entropy serves as the loss function during training.

5.3 Performance metrics

Network intrusion detection systems are measured using nine primary evaluation metrics: accuracy, precision, recall, detection rate (DR), false positive rate (FPR), F1 score, G-mean, the receiver operating characteristic (ROC) curve, and the area under the ROC curve (AUC). These performance indicators are computed using values obtained from the confusion matrix during the classification of network traffic.

5.4 Results and Discussion

Multiple experiments are conducted to assess the effectiveness of the proposed GAVAE-DNN model in detecting previously unseen attacks and handling class imbalance issues. Tables.4 and 5 display the distribution of the newly synthesized samples across different categories within the training sets of the UNSW-NB15 and NSL-KDD datasets, respectively.

Figs.8 and 9 illustrate the spatial distribution of both the original and augmented training samples, projected into two dimensions using UMAP (Uniform Manifold Approximation and Projection). Furthermore, the detection capability of the proposed GAVAE-DNN model is compared against several state-of-the-art intrusion detection approaches from existing literature, with the comparative results presented in Tables 6 and 7.

Table 4. Generated Training Samples for the NSL-KDD Dataset

Category	Number of actual records	Number of newly generated records	Total
Normal	13,449	0	13,449
Probe	2,289	11,160	13,449
DoS	9,234	4,215	13,449
U2R	11	13,438	13,449
R2L	209	13,240	13,449
Total	**25,192**	**42,053**	**67,245**

Table 5. Generated Training Samples for the UNSW-NB15 dataset

Category	Number of actual records	Number of newly generated records	Total
Normal	56,000	0	56,000
Generic	40,000	16,000	56,000
Exploits	33,393	22,607	56,000
Fuzzers	18,184	37,816	56,000
DoS	12,264	43,736	56,000
Reconnaissance	10,491	45,509	56,000
Analysis	2,000	54,000	56,000
Backdoor	1,746	54,254	56,000
Shellcode	1,133	54,867	56,000
Worms	130	55,870	56,000
Total	**175,341**	**481,340**	**560,000**

Figs.8 and 9 present a two-dimensional projection of the spatial distribution of the original and newly generated training samples, visualized using Uniform Manifold Approximation and Projection (UMAP). Additionally, GAVAE- DNN's detection performance is contrasted with that of other cutting-edge models documented in the IDS literature, and the comparative outcomes are displayed in Tables 6 and 7.

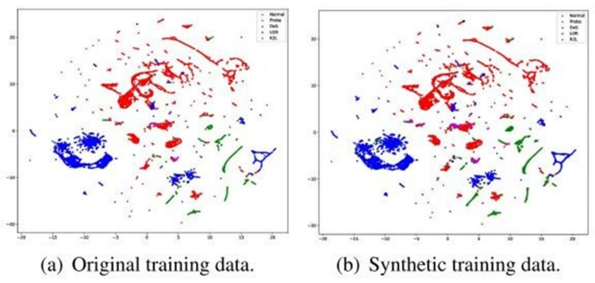

Fig. 8. UMAP visualization for NSL-KDD dataset

5.5 Comparative Analysis with other state-of-the-art methods

To enhance the detection of previously unseen attacks, this research presents the GAVAE-DNN model, which generates synthetic unknown attack samples and addresses class imbalance within the training data. The model uses a decoder to produce additional samples for underrepresented attack categories. In addition, five widely adopted classification techniques—K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Random Forest (RF), Convolutional Neural Network (CNN), and Deep Neural Network (DNN)—are employed to perform intrusion detection. Their performance is assessed and compared with the proposed GAVAE-DNN model to determine its effectiveness. These classification methods have been widely applied in intrusion detection studies. The comparative experimental results are presented in Figs.10 and 11.

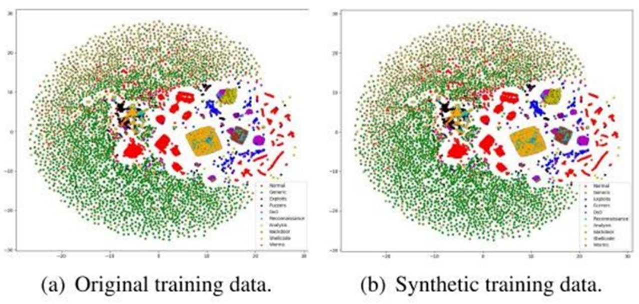

Fig. 9. UMAP visualization UNSW-NB15 dataset

Table 6. Performance Comparison of Classifiers on NSL-KDD Dataset

Method	Accuracy	Recall (DR)	Precision	F1	G-mean	FPR
KNN	76.83	63.26	63.76	75.43	77.54	7.10
CNN	73.54	57.34	72.36	70.64	73.52	7.18
DNN	75.87	61.23	83.56	74.98	76.23	6.68
RF	78.26	66.98	91.54	77.14	79.34	6.77
SVM	81.34	71.36	93.54	81.36	81.33	7.41
GAVAE-DNN	90.35	85.76	96.04	91.55	90.08	4.65

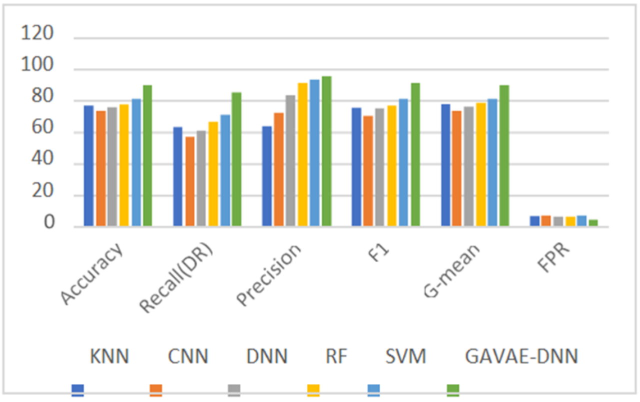

Fig. 10. Performance Comparison of Classifiers on NSL-KDD Dataset

Table 7. Performance Comparison of Classifiers on UNSW-NB 15 dataset

Method	Accuracy	Recall (DR)	Precision	F1	G-Mean	FPR
KNN	86.74	84.65	83.03	87.54	84.75	26.44
CNN	48.08	51.85	53.64	52.38	48.32	56.43
DNN	86.34	92.43	83.56	88.34	86.04	24.89
RF	88.23	91.45	83.56	90.43	86.32	20.44
SVM	87.34	93.63	82.65	90.13	85.43	25.77
GAVAE-DNN	93.01	94.56	95.32	93.88	93.12	5.67

Fig. 10 shows that the GAVAE-DNN model outperforms other classifiers on the NSL-KDD dataset, delivering higher scores in accuracy, recall, precision, F1 score, and G-mean. The Fig. 10 compares the performance of six classification models—KNN, CNN, DNN, RF, SVM, and GAVAE-DNN—using all the key evaluation metrics. Among these, GAVAE-DNN consistently achieves the highest values across all positive metrics while maintaining the lowest FPR, indicating superior detection capabilities. SVM and RF follow closely with balanced performance, whereas CNN underperforms across most categories. This visual clearly demonstrates the advantage of combining generative and discriminative learning methods. Thus GAVAE-DNN improves the accuracy and low false alarms.

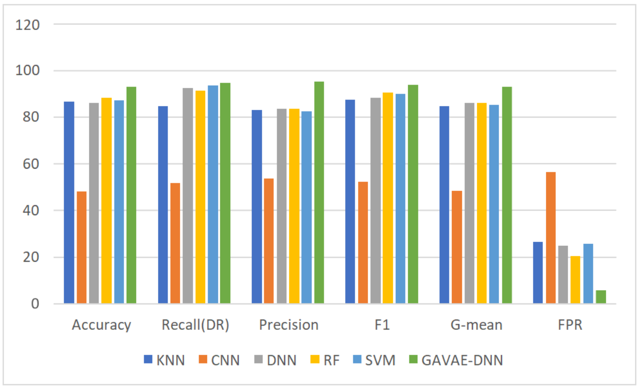

Fig. 11. Performance Comparison of Classifiers on UNSW-NB 15 dataset

Likewise, Fig. 11 demonstrates that the GAVAE-DNN model delivers superior results on the UNSW-NB15 dataset, with improved overall accuracy, precision, F1 score, G-mean, and a reduced false positive rate (FPR). The Fig. 11 presents a comparison of six models—KNN, CNN, DNN, RF, SVM, and GAVAE-DNN—using key performance metrics. Among them, GAVAE-DNN consistently achieves the highest scores across all the metrics. It also records the lowest False Positive Rate (FPR), making it the most reliable and accurate. In contrast, CNN lags behind significantly, particularly in Recall and FPR. The results highlight GAVAE-DNN's capability to deliver superior detection with minimal errors.
These comparative experimental results have shown the significance of the proposed GAVAE-DNN model in detecting network intrusions, especially when dealing with imbalanced network traffic.

6. CONCLUSION

The need for network intrusion detection is growing as network intrusion continues to change. Imbalanced network traffic presents a major threat to cybersecurity, complicating the ability of intrusion detection systems to effectively identify malicious activities. This research introduces the Generative Adversarial Variational Autoencoder (GAVAE) as a method to improve classification models' ability to learn from skewed network data. The GAVAE framework generates a broad range of rare and previously unseen attack samples. To enhance training data diversity and maintain balance, its decoder is employed to create supplementary attack samples for designated labels. The hidden layer weights of the DNN are initialized using the encoder from GAVAE, which simultaneously performs automatic extraction of high-level feature representations from the original data samples. The proposed hybrid architecture for network intrusion detection, called GAVAE-DNN, combines GAVAE with DNN. The effectiveness of the GAVAE-DNN model was assessed using the NSL-KDD and UNSW-NB15 benchmark datasets, producing favorable outcomes. Comprehensive experimental evaluations reveal that GAVAE-DNN excels in detecting both known and novel attacks while also improving the detection rate for infrequent attacks. Additionally, comparative studies on the UNSW-NB15 dataset highlight its strong capability in identifying advanced network threats. The model attained a maximum accuracy of 93.01 % and an F1 score of 93.88 % on the dataset. Given its extremely competitive findings when compared to the most advanced models, the suggested SAVAER could be a competitive option for network intrusion detection.

7. REFERENCES

[1] Ankalaki, S., Rajesh, A. A., Pallavi, M., Hukkeri, G. S., Jan, T., Naik, G. R. (2025). Cyber Attack Prediction: From Traditional Machine Learning to Generative Artificial Intelligence. IEEE Access..

[2] Liu, Chang, Ruslan Antypenko, Iryna Sushko, and Oksana Zakharchenko. ”Intrusion detection system after data augmentation schemes based on the VAE and CVAE.” IEEE Transactions on Reliability 71, no. 2 (2022): 1000-1010.

[3] Zavrak, Sultan, and Murat Iskefiyeli. ”Anomaly-based intrusion detection from network flow features using variational autoencoder.” IEEE Access 8 (2020): 108346- 108358.

[4] Liu, Lan, Pengcheng Wang, Jun Lin, and Langzhou Liu. ”Intrusion detection of imbalanced network traffic based on machine learning and deep learning.” IEEE access 9 (2020): 7550-7563.

[5] Zhang, Xiaoxuan, Jing Ran, and Jize Mi. ”An intrusion detection system based on convolutional neural network for imbalanced network traffic.” In 2019 IEEE 7th international conference on computer science and network technology (ICCSNT), pp. 456-460. IEEE, 2019.

[6] Liu, Hongyu, and Bo Lang. ”Machine learning and deep learning methods for intrusion detection systems: A survey.” applied sciences 9, no. 20 (2019): 4396.

[7] Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... Bengio, Y. (2014). Generative adversarial networks. arXiv 2014. arXiv preprint arXiv:1406.2661, 10

[8] Bamhdi, Alwi M., Iram Abrar, and Faheem Masoodi. ”An ensemble based approach for effective intrusion detection using majority voting.” Telkomnika (Telecommunication Computing Electronics and Control) 19, no. 2 (2021): 664-671.

[9] Bhati, Nitesh Singh, and Manju Khari. ”A new ensemble based approach for intrusion detection system using voting.” Journal of Intelligent Fuzzy Systems 42, no. 2 (2022): 969-979.

[10] Abbas, Adeel, Muazzam A. Khan, Shahid Latif, Maria Ajaz, Awais Aziz Shah, and Jawad Ahmad. ”A new ensemble-based intrusion detection system for internet of things.” Arabian Journal for Science and Engineering (2022): 1-15.

[11] Shu, Dule, Nandi O. Leslie, Charles A. Kamhoua, and Conrad S. Tucker. ”Generative adversarial attacks against intrusion detection systems using active learning.” In Proceedings of the 2nd ACM workshop on wireless security and machine learning, pp. 1-6. 2020.

[12] Gan, Baiqiang, Yuqiang Chen, Qiuping Dong, Jianlan Guo, and Rongxia Wang. ”A convolutional neural network intrusion detection method based on data imbalance.” The Journal of Supercomputing 78, no. 18 (2022): 19401-19434.

[13] Arafah, Mohammad. ”Improving intrusion detection system performance using generative adversarial networks architecture.” PhD diss., Loughborough University, 2023.

[14] Vu, Ly, Quang Uy Nguyen, Diep N. Nguyen, Dinh Thai Hoang, and Eryk Dutkiewicz. ”Deep generative learning models for cloud intrusion detection systems.” IEEE Transactions on Cybernetics 53, no. 1 (2022): 565-577.

[15] Ibrahim, Bekkouch Imad, Dragoş Constantin Nicolae, Adil Khan, Syed Imran Ali, and Asad Khattak. ”VAE-GAN based zero-shot outlier detection.” In Proceedings of the 2020 4th international symposium on computer science and intelligent control, pp. 1-5. 2020.

[16] Zhang, Ying, and Qiang Liu. ”On IoT intrusion detection based on data augmentation for enhancing learning on unbalanced samples.” Future Generation Computer Systems 133 (2022): 213-227.

[17] Zhang, Junjie, and Ying Zhao. ”Research on intrusion detection method based on generative adversarial network.” In 2021 International Conference on Big Data Analysis and Computer Science (BDACS), pp. 264-268. IEEE, 2021.

[18] Navya, V. K., J. Adithi, Diksha Rudrawal, Harshita Tailor, and Nileena James. ”Intrusion detection system using deep neural networks (DNN).” In 2021 International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA), pp. 1-6. IEEE, 2021.

[19] Samriya, Jitendra Kumar, Rajeev Tiwari, Xiaochun Cheng, Rahul Kumar Singh, Achyut Shankar, and Manoj Kumar. ”Network intrusion detection using ACO-DNN model with DVFS based energy optimization in cloud framework.” Sustainable Computing: Informatics and Systems 35 (2022): 100746.

[20] Zhao, C., Du, H., Niyato, D., Kang, J., Xiong, Z., Kim, D. I., ... Letaief, K. B. (2024). Generative AI for secure physical layer communications: A survey. IEEE Transactions on Cognitive Communications and Networking.

[21] Liu, Zhiqiang, Mohi-Ud-Din Ghulam, Ye Zhu, Xuanlin Yan, Lifang Wang, Zejun Jiang, and Jianchao Luo. ”Deep learning approach for IDS: using DNN for network anomaly detection.” In Fourth International Congress on Information and Communication Technology: ICICT 2019, London, Volume 1, pp. 471-479. Springer Singapore, 2020.

[22] Vu, T. H., Jagatheesaperumal, S. K., Nguyen, M. D., Van Huynh, N., Kim, S., Pham, Q. V. (2024). Applications of Generative AI (GAI) for Mobile and Wireless Networking: A Survey. arXiv preprint arXiv:2405.20024.

[23] Moraboena, Srikanthyadav, Gayatri Ketepalli, and Padmaja Ragam. ”A Deep Learning Approach to Network Intrusion Detection Using Deep Autoencoder.” Revue d’Intelligence Artificielle 34, no. 4 (2020).

[24] Andresini, Giuseppina, Annalisa Appice, and Donato Malerba. ”Autoencoder- based deep metric learning for network intrusion detection.” Information Sciences 569 (2021): 706-727.

[25] Sadaf, Kishwar, and Jabeen Sultana. ”Intrusion detection based on autoencoder and isolation forest in fog computing.” IEEE Access 8 (2020): 167059-167068.

[26] Yang, Y., Zheng, K., Wu, B., Yang, Y., Wang, X. (2020). Network intrusion detection based on supervised adversarial variational auto-encoder with regularization. IEEE access, 8, 42169-42184.