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

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 increasingly complicated with the arrival of new Internet technologies such as file sharing, mobile payments, and the expansion of the Internet of Things [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 security mechanism that helps protect computers and network systems from possible abuse is intrusion detection [3]. The anomaly detection approach, which compares anomalous behavior with normal behaviour, and the signature-based detection method, which compares with pre-configured and pre-determined attack patterns known as signatures, are the two ways intrusion detection systems identify intrusions. In an effort to reduce false detection of unknown attacks and increase the identification of preconfigured attacks, hybrid detection - which combines anomaly-based and signature-based detection - has recently been the subject of numerous studies [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.

Later, the researchers used machine learning techniques to detect intrusions [1]. However, machine learning was not well received at the time because of the limitations in computer power and storage. 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 address the challenges in the existing system, the proposed work creates a generative adversarial network (GAN)-assisted intrusion detection system (IDS) that may mitigate the dataset-related restrictions for all of these developing 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]. Two neural networks are combined to form GAN; one creates the objects, while the other estimates them. G stands for generator in the first network, and D stands for discriminator in the second. They are in competition with one another: D learns to identify these fakes from the data, while G learns to produce ever-more-believable things. The direct route across the network for information transmission from G to D is called the channel, and the reverse path from D to G is known as the return path. Giving the network a thorough grasp of the hidden data and its contents is the primary objective of all neural models based on GANs. This functionality will assist us in resolving the lack of dataset problem.

When compared to typical ANNs, GAN offers several noteworthy benefits, including the ability to quickly search for and classify network anomalies as well as generate extra anomalous copies to enhance the quality of the flagged samples. Since there is no need to generate each entry in the sample sequentially, GAN generates samples faster than even fully visible belief nets (Neural Autoregressive Distribution Estimation (NADE), Pixel Recurrent Neural Network (RNN), WaveNet, etc.). Unlike Boltzmann machines, which rely on a Monte Carlo approximation to the gradient of the log partition function, GANs train without the need for any such approximations and are also significantly simpler to train. GAN does not introduce deterministic bias in contrast to variation autoencoders.

There is a high degree of category imbalance because most traffic data in real cyberspace is from normal activities, with very few being harmful cyberattacks. Normal activities hold the dominant position in cyberspace. In the incredibly redundant and unbalanced network traffic data, intrusion detection is under a lot of strain. Cyberattacks might blend in with a lot of everyday traffic. Because of this, it is simple to misclassify and the machine learning algorithm is unable to adequately understand the distribution of a small number of categories [5]. Adaptive learning of the difference between normal and abnormal behavior is a powerful tool for improving the real-time intrusion processing performance by training a large number of data samples. Nonetheless, the imbalance in classification still has an impact on the multiclassification of network traffic. In this work a unique Generative Adversarial Variational Autoencoder (GAVAE) approach is proposed to address the class imbalance issue in network traffic when faced with imbalanced data. This method effectively addresses data imbalance and strengthens the classification model's ability to learn difficult examples. To evaluate its effectiveness, two benchmark datasets are tested using both deep learning and traditional machine learning algorithms. The main contributions of this work are as follows.

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

—In order to address the issue of class imbalance in intrusion detection and improve the classifier's ability to learn distinctions during training, this work presents a novel GAVAE technique that reduces the majority samples and augments the minority samples in the challenging set.

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 Intrusion Detection Systems (HIDS) and Network-Based Intrusion Detection Systems (NIDS). HIDS is employed by network administrators to track and assess activities on a particular 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, 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]. In this article, three data-based research schemes—a data augmentation scheme based on the variational autoencoder (VAE), a data-balancing strategy based on the conditional VAE, and a data-balancing scheme based on both conditional VAE and random under sampling—are built step-by-step. 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.

In order to detect intrusions in unbalanced network traffic, [4] investigates deep learning and machine learning techniques. A novel approach, known as the Difficult Set Sampling Technique (DSSTE), is proposed to tackle the challenge of class imbalance. Additionally, a Convolutional Neural Network (CNN)-based intrusion detection model is introduced in [5]. Before training the CNN, network traffic is balanced using the Synthetic Minority Oversampling Technique and Edited Nearest Neighbors (SMOTE-ENN) method. The model's performance is evaluated using the NSL-KDD dataset.

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. They chose the Kyoto2006+ dataset, which is older and more promising than the KDDCup '99 dataset, which is the most employable. This forces them to strive for a specific degree of accuracy, although in a few instances, the recall of the result is rather low, indicating high false-negative rate (FPR) levels.

In order to address the problems with single classifiers, an ensemble approach has been proposed in [8]. As a result, a highly scalable and constructive majority voting-based ensemble model was proposed, which can be used in real-time to successfully examine network traffic and proactively warn about potential attacks. 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. Traditional machine learning techniques struggle with data imbalance and feature redundancy when dealing with complicated and vast network data feature information. This leads to low detection rates, high false alarm rates, and subpar real-time performance of intrusion detection systems. Thus [12] presents a data imbalance-based Convolutional Neural Network Intrusion Detection Method (CNN-IDMDI) to address these issues. 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. Many studies, such as Multilayer Percetron [15], Convolutional Neural Network [16], and Recurrent Neural Network [17], used neural networks for intrusion detection, motivated by the amazing impact of deep learning. In addition, Aljawarneh et al. [18] reported an enhanced accuracy intrusion detection system based on feature selection and hybrid algorithm.

The efficacy of most intrusion detection systems is hampered by class-imbalanced data, despite the significant advancements gained in previous approaches [19]. The issue of class imbalance arises when there are substantially less intrusion samples than the typical amount. Deep learning technologies, which can produce results that are on the same level with or even better than those of human experts, are widely used in computer vision, natural language processing, speech recognition, drug design, intrusion detection, and other fields. Examples of these technologies include DBN (Deep Belief Network), DNN (Deep Neural Network), CNN (Convolutional Neural Network), LSTM (Long-Term Short-Term Memory), and GAN (Generate Adversarial Network) [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 the quick expansion of network traffic data.

On unbalanced data sets, it is challenging for conventional classifiers to reach high detection rates. Second, the security of the Internet and Intranet is increasingly being threatened by unidentified assaults as a result of the widespread use of new technologies like artificial intelligence (AI). Conventional classifiers identify unknown assaults poorly, while they perform better on known threats. Third, the quantity and complexity of network traffic data are increasing due to the growing acceptance of the Internet of Things and the broad use of cloud- based services, making it challenging for conventional classifiers to discern between normal and aberrant activity.

In conclusion, even though the previously described deep learning techniques have shown good results for network intrusion detection systems, they are still plagued by low detection rates for unknown and infrequent attacks. In order to address these issues, developing a novel hybrid intrusion detection system is proposed. The suggested framework concatenates the encoded latent vectors with the given attack labels and feeds them into the decoder to produce unknown assaults. GAVAE is used to learn the latent distribution of the intrusion data. As a result, the training sample set is balanced and the diversity of training samples is enhanced. In order to create a DNN classifier, the GAVAE encoder also includes a softmax layer. Lastly, the DNN classifier efficiently identifies unknown

threats by automatically examining high- level abstract feature representation of network attack data.

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 encoder and a decoder make up the two main parts of an autoencoder's architecture as shown in Fig 1. While the decoder is a decompression technique that can be used to generate new data or denoise raw data, the encoder is typically employed to reduce dimensionality. Mapping the given raw input data (x) into the latent space of representation is the encoder's job is given in the Eq.1.

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

where W and b stand for the weight matrix and bias vector, respectively, and f signifies the encoder's activation function. 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., $\tilde{x}$).

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

where g stands for the decoder's activation function and W' and b' stand for the weight matrix and bias vector, respectively. As a result, training the autoencoder to reduce reconstruction error L_{RE} as shown in Eq.3.

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

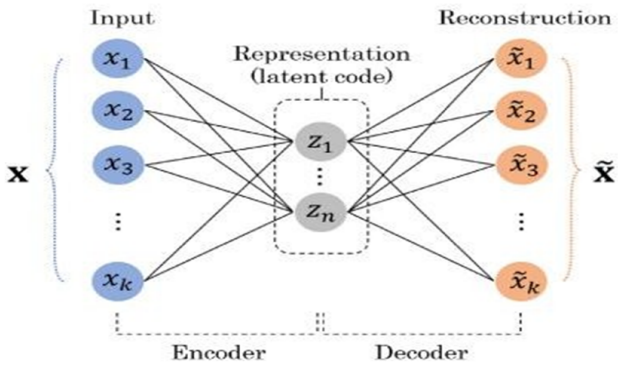


Fig. 1. Auto-Encoder architecture

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

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]. 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.

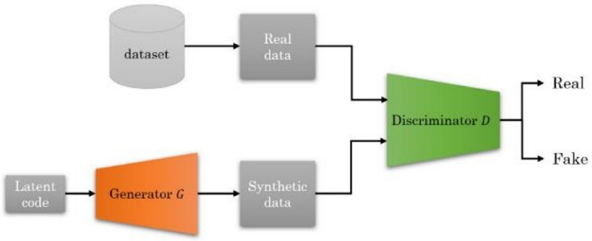


Fig. 2. Architecture of Generative adversarial Network

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:

$$V(D,G) = \min_G \max_D E_{\mathbf{x}\sim p_{data}}[\log D_{\theta_D}(\mathbf{x})] + E_{\mathbf{z}\sim p_z}[\log(1 - D_{\theta_D}(G_\theta \tag{4}$$

where the model parameters of D and G are indicated by $\theta_{\mathbf{D}}$ and $\theta_{\mathbf{G}}$ respectively. As a result, the generator is trained to provide fake data that can maximize the discriminator's confidence score, and the discriminator is trained to output a greater confidence value in actual data. When this training process is repeated enough times, the discriminator and generator will reach a point where there is no more room for improvement.

3.3 Generative Adversarial Auto- Encoder

Generative Adversarial Auto-Encoder (GAAE) is illustrated in Fig. 3. Because AAE and VAE act similarly, AE's latent variable z is compelled to follow a predetermined prior distribution p(z). When it comes to the AAE, z can be defined in any way and is simple to sample and feed into the discriminator. But unlike VAE, which maximizes the evidence lower bound (ELBO), AAE uses the GAN to direct the encoder $q(z \mid x)$ distribution so that it matches the previous distribution p(z). The discriminator network is trained to distinguish between real data from the previous distribution p(z) and false data from the AE in the input code vector z.
The adversarial auto-encoder with regularization integrates label information throughout the adversarial training phase. To give the latent variables a better fit distribution, the discriminator network receives the one-hot label. As demonstrated, the decoder's input of one- hot label converts unsupervised training to supervised training. GAAE joins the hidden variable z with the one-hot label y

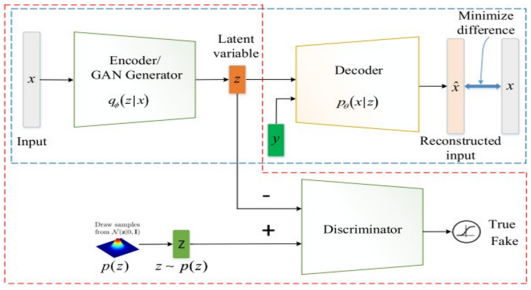


Fig. 3. Adversarial Auto-Encoder Framework

collectively as the decoder's input. Next the network is compelled to acquire the 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.

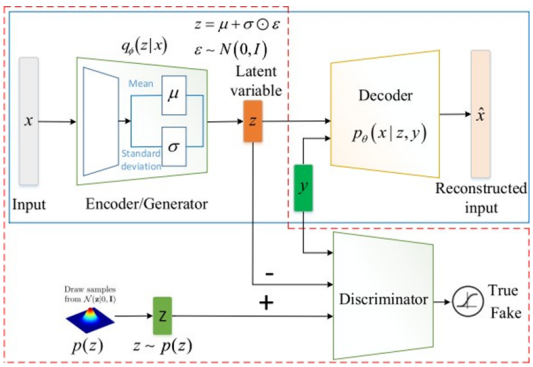


Fig. 4. Proposed GAVAE architecture

Additionally, the one-hot class vectors are given to the discriminator and decoder in order to control the degree of independence between latent variables and classes. The distribution ($q(x \mid z)$, y) can provide attack samples of the given label y to the decoder. The 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. During the outlier analysis stage, the system removes anomalies that could have an adverse effect on the training of the model. 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:

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

(5)

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. A binary vector the size of the number of attribute values is used to represent each nominal (categorical) attribute, with 1 given to a point that corresponds to the expressed value and 0 to all other points. 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 [28] and standardization [24]. Between the two strategies, the min-max normalization strategy 2 wasselected. For a numeric attribute A, the normalization function (f_A (.) that maps$\forall x \in Aintoarange[0,1]isdefined(Eq.6)asfollows$:

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

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 process for the proposed GAVAE follows these steps for each minibatch: The Variational Autoencoder (VAE) is trained to minimize the binary cross-entropy loss by reducing the difference between the reconstructed data $\hat{x}$ and the original input data x.

—(1) The discriminator learns to differentiate between real and generated samples z, distinguishing those from the real data distribution q(z) and the multivariate Gaussian prior p(z), with the objective of minimizing the WGAN-GP loss.

—(2) The generator (encoder) is optimized to deceive the discriminator by generating samples z that closely resemble real data.

After training SAVAER, the original training dataset (x,y) is passed through the trained VAE to compute the cross-entropy reconstruction loss. The binary cross-entropy reconstruction loss for a given sample (x_i,y_i) is defined (Eq.7) as follows:

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

where $\hat{x}_i$ = Decoder (Encoder(x_i,y_i)); x_i ,k represents the k-th feature value of the sample x_i ; d denotes the number of features. The maximum class reconstruction loss maxLj of class j is defined in the Eq.8 as follows:

$$\max L_j = k \cdot \max\{L(x_i, y_i) \mid y_i \in classj\} \quad (8)$$

where k is the scale factor, and usually k 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}.

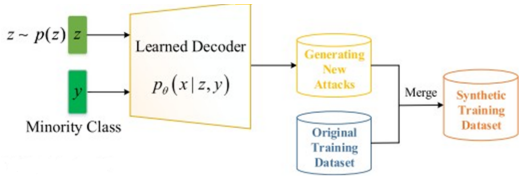


Fig. 5. Data Augmentation

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 ensures that the newly generated samples have the same spatial distribution as the original samples. In order to balance the training data set, the freshly created attack samples that are consistent with the original data distribution are finally combined with the original training data set.

$$S = \{ S \cup \{x_{new}, y_{new}\}, if y_{new} \in classj \quad \& L(x_{new}, y_{new}) \leq \max L_j \} \quad (9)$$

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 synthetic training data set to refine the classifier.

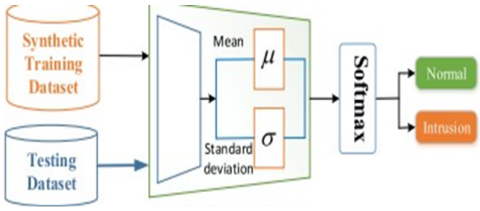


Fig. 6. Detecting attacks

5. EXPERIMENT AND DISCUSSION

All experiments are conducted on a ThinkStation workstation equipped with an Intel E5-2620 CPU and 64 GB RAM, running a 64-bit Windows 10 operating system within a TensorFlow environment. To evaluate the performance of the proposed model and several widely used classifiers, three distinct datasets are utilized: NSL-KDD (KDDTest+)[24], NSL-KDD (KDDTest-21) [24], and UNSW-NB15 [25]. The optimal hyperparameters and network architecture for the proposed model are determined using grid search and 5-fold cross-validation, ensuring the highest prediction accuracy.

5.1 Dataset description

For evaluation of any intrusion detection technique, data collection is needed. The NSL-KDD dataset [24] and the UNSW-NB15 dataset [25] are adopted to evaluate GAVAE - IDS.

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	Basic features of network connections
10–22	Content-related traffic features
23–31	Time-related traffic features
32–41	Host-based traffic 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): A Distributed Denial-of-Service (DDOS) assault, such as Smurf, Ping of Death, Neptune, etc., is when an attacker prevents memory resources or other network resources from responding to proper requests, or overloads them.

(2) Remote 2 User (R 2 L): R 2 L 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 (U 2 R): 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 was presented by Moustafa et al. [19] in 2015 and is a combination of different anomalies. The many forms of attacks in UNSW-NB15 provide a good opportunity to test various intrusion detection systems. Table 2 lists 42 properties for each sample in UNSW-NB15, including Basic characteristics, content-related characteristics, time-related characteristics, general purpose characteristics, and connection-based characteristics. Nonetheless, UNSW-NB15 and NSL-KDD are examples of common imbalanced datasets. As indicated in Table 3, the NSL-KDD imbalance ratios range from 1.44 to 305.77, while the UNSW-NB15 imbalance ratios range from 10.30 to 12,751.50. Furthermore, in the NSLKDD test set, 38 anomaly species were found, compared to 23 species in the training set.

Table 2. Description of the UNSW-NB15 attributes.

Attributes	Description
1–13	Basic features of network connections
14–21	Content-related traffic features
22–30	Time-related traffic features
31–35	General purpose traffic features
36–42	Connection-based traffic features

Instead of binary classification, a 5-class classification is applied to the NSL-KDD dataset and a 10-class classification to the UNSW-NB15 dataset. The intrusion detection task is approached as a multiclass classification problem using multiple One-vs-All schemes. In each scheme, one specific class is treated as the positive class, while the remaining classes are considered negative.

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

In order to address the class imbalance in NSL-KDD, samples for the four minority classes DoS, Probe, R2L, and U2R are generated using the GAVAE module. 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.

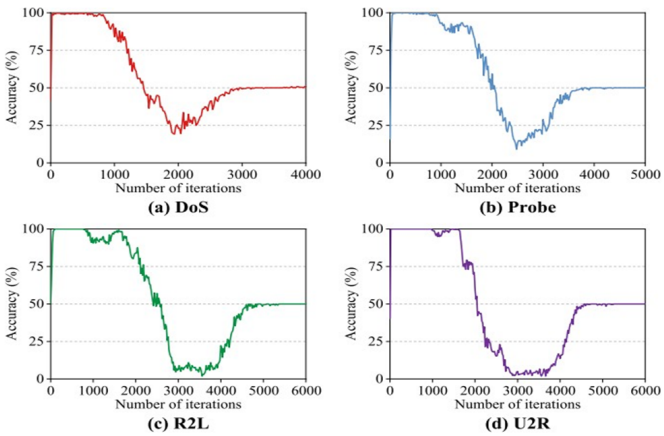


Fig. 7. Convergence curve of discriminator D

Using the curve of DoS Fig. 7(a)) as an example, D was able to distinguish the synthetic samples from the genuine ones with ease during the first 1000 iterations since G was unable to produce representative samples. 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 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. To optimize the DNN, the Adam algorithm is employed with a learning rate of 0.00005, and Categorical Cross-Entropy is used as the loss function.

5.3 Performance metrics

Network intrusion detection systems are evaluated using nine key performance metrics: accuracy, precision, recall, detection rate

(DR), false positive rate (FPR), F1 score, G-mean, receiver operating characteristic (ROC) curve, and area under the ROC curve (AUC). These metrics are derived from the confusion matrix when categorizing network attacks.

—True Positive (TP): The number of correctly detected attack instances.

—True Negative (TN): The number of correctly classified normal traffic records.

—False Positive (FP): The count of normal traffic records incorrectly identified as attacks.

—False Negative (FN): The number of attack instances mistakenly classified as normal traffic.

5.3.1 Accuracy. The percentage of accurately anticipated attacks and normal records to the total number of records is known as the accuracy. A greater accuracy indicates better classification model performance ($accuracy \in [0, 1]$).

5.3.2 Detection rate. The percentage of real assault reports that are accurately identified is known as the recall rate, or DR (detection rate). DR is sometimes referred to as Sensitivity or TPR (true positive rate). The classification model performs better if the DR is higher ($DR \in [0, 1]$).

5.3.3 Specificity. The percentage of normal records that are accurately identified is measured by the TNR (true negative rate), also known as specificity. A greater TNR indicates better classification model performance ($TNR \in [0, 1]$).

5.3.4 Precision. The percentage of all predicted attack records that are actual attack records is known as the precision. A greater Precision indicates better classification model performance ($Precision \in [0, 1]$).

5.3.5 False positive rate. The percentage of regular records that are mistakenly identified as attack records is known as the false positive rate, or FPR. The classification model performs better if the FPR is lower ($FPR \in [0, 1]$).

5.3.6 F1 Score. The harmonic mean of recall and precision is known as the F1 score. When it comes to assessing the effectiveness of the classification model in unbalanced data sets, the F1 score is far more useful than accuracy. A greater F1 indicates better classification model performance ($F1 \in [0, 1]$).

5.3.7 G-mean. The balance between majority and minority categorization performance is measured by the G - mean, which is the geometric mean of specificity and sensitivity. The classification model performs better if the G mean is higher ($G - mean \in [0, 1]$).

5.3.8 ROC Curve. The receiver operating characteristic curve or ROC is a widely used graph that shows a classifier's performance across all potential thresholds. A curve that is produced on a two-dimensional plane that shows the ordinate of the TPR and the abscissa of the FPR when the threshold for classifying data is changed is the primary analytical tool.

5.3.9 AUC. The area under the ROC curve is called the area under the curve, or AUC for short. A larger AUC indicates better classification model performance ($AUC \in [0, 1]$). Real classifiers typically have an AUC of 0.5 to 1.

5.4 Results and Discussion

Several experiments are performed to evaluate the efficiency of the proposed GAVAE-DNN in identifying unknown attacks and addressing data imbalance. The distribution of newly generated samples for each category in the UNSW-NB15 and NSL-KDD training datasets is shown in Tables.4 and 5.

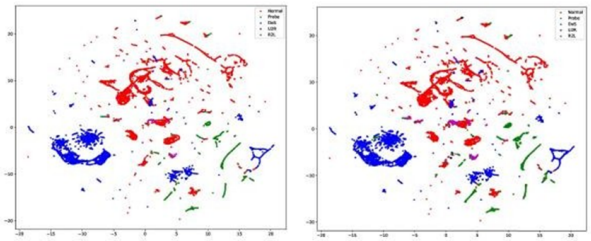
Table 4. Training samples generated for NSL-KDD

Category	Number of original 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. Training samples generated for UNSW-NB15

Category	Number of original 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**

The spatial distribution of the sample of the original and enhanced training data, projected onto a two-dimensional space using UMAP (Uniform Manifold Approximation and Projection), is shown in Figs.8 and 9. 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.

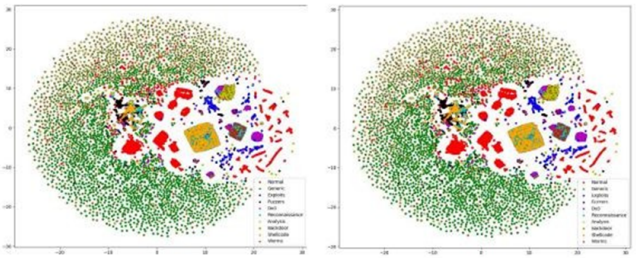


(a) Original training data. (b) Synthetic training data.

Fig. 8. UMAP visualization for NSL-KDD dataset

5.5 Comparative study with other state-of -art methods

To improve the detection rate of unknown attacks, this study introduces the GAVAE-DNN model, which generates unknown attack instances and balances the training dataset. The model employs a decoder to create minority attack samples for training. Furthermore, five commonly used classification models—K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Random Forest (RF), Convolutional Neural Network (CNN), and Deep Neural



(a) Original training data. (b) Synthetic training data.

Fig. 9. UMAP visualization UNSW-NB15 dataset

Network (DNN)—are utilized for 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.

Table 6. Comparison of overall performance of different classification models 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

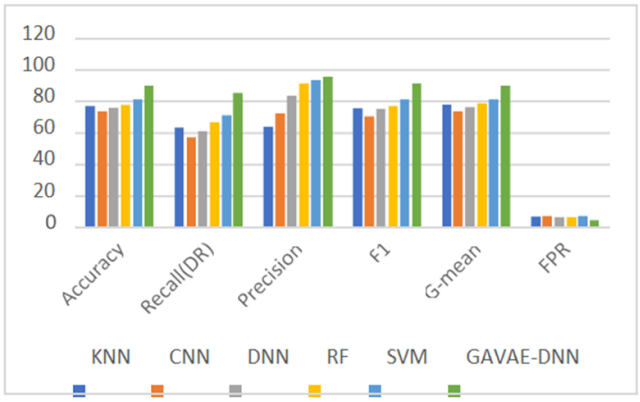


Fig. 10. Comparison of overall performance of different classification models on NSL-KDD dataset

Table 7. Comparison of overall performance of different classification models 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

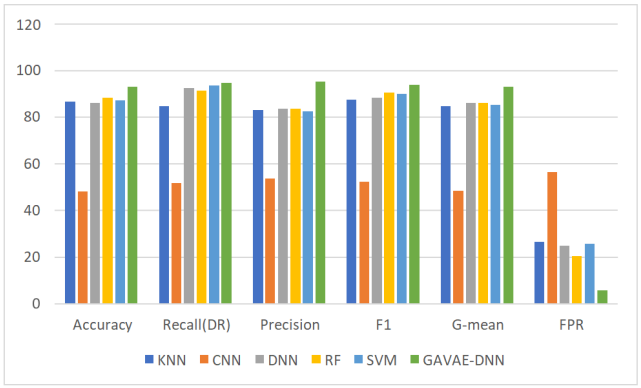


Fig. 11. Comparison of overall performance of different classification models on UNSW-NB 15 dataset

Fig. 10 demonstrates that the GAVAE-DNN model achieves superior performance compared to other classification models on the NSL-KDD dataset, excelling in overall accuracy, recall, precision, F1 score, and G-mean. Fig. 10 illustrates the comparative performance of six classification models—KNN, CNN, DNN, RF, SVM, and GAVAE-DNN—using metrics such as Accuracy, Recall, Precision, F1-score, G-mean, and False Positive Rate (FPR). 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.
Likewise, Fig. 11 highlights that GAVAE-DNN outperforms other models on the UNSW-NB15 dataset, achieving higher overall accuracy, precision, F1 score, G-mean, and a lower false positive rate (FPR). Fig. 11 compares six models—KNN, CNN, DNN, RF, SVM, and GAVAE-DNN—across key evaluation metrics. GAVAE-DNN consistently performs best, scoring the highest in Accuracy, Recall, Precision, F1, and G-mean. 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 findings confirm the effectiveness 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 weights of the DNN's hidden layers are initialized using GAVAE's encoder, which also automatically extracts high-level feature representations from the original 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 achieved a peak accuracy of 93.01 % and an F1 score of 93.88 % on this 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.