Computers & Security 125 (2023) 103054

Contents lists available at ScienceDirect

Computers & Security

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

Synthetic attack data generation model applying generative adversarial network for intrusion detection

Vikash Kumar[a,b,*], Ditipriya Sinha[b]

[a] *Department of Computer Science & Engineering, Siksha O Anusandhan Deemed to be University, Bhubaneswar, India*
[b] *Department of Computer Science & Engineering, National Institute of Technology Patna, Patna, Bihar, India*

ARTICLE INFO

Article history:
Received 15 November 2021
Revised 21 November 2022
Accepted 5 December 2022
Available online 7 December 2022

Keywords:
Intrusion detection system
Cyber-attack
Generative adversarial networks
Data synthetization
Data imbalance

ABSTRACT

Detecting a large number of attack classes accurately applying machine learning (ML) and deep learning (DL) techniques depends on the number of representative samples available for each attack class. In most cases, the data samples are highly imbalanced that results in a biased intrusion detection model towards the majority classes. Under-sampling, over-sampling and SMOTE are some techniques among the solutions that turn the imbalanced dataset to balanced one. These techniques have not had much impact on the improvement of detection accuracy. To deal with this problem, this paper proposes a Wasserstein Conditional Generative Adversarial Network (WCGAN) combined with an XGBoost Classifier. Gradient penalty along with the WCGAN is used for stable learning of the model. The proposed model is evaluated with some other GAN models (i.e., standard/vanilla GAN, Conditional GAN) which shows the significance of applying WCGAN in this paper. The loss on generated and real data shows a similar pattern and is lower for the Wasserstein variants of GAN compared to the other variants of the GAN model. The performance is benchmarked on three datasets NSL-KDD, UNSW-NB15 and BoT-IoT. The comparison of performance metrics before and after using the proposed framework with XGBoost classifier shows improvement in terms of higher precision, recall and F-1 score. However, comparatively less improvement is observed in FAR compared to other classifiers such as Random Forest (RF), Decision Tree (DT), Support Vector Machine (SVM). The proposed work is also compared with a recent similar technique called DGM, which uses conditional GAN along with different ML classification models. The performance of the proposed model outperforms DGM. The proposed model creates a significant footprint (or, attack signatures) to tackle with the problem of data-imbalance during the design of the Intrusion Detection System (IDS).

1. Introduction

Over the past few years, technologies like the Internet of Things (IoT), Cloud Computing, and Artificial Intelligence (AI) have gained popularity in the mainstream Internet. As a result, a huge number of devices are introduced to the Internet and the amount of data generated by these devices increasing tremendously day by day. The rapid growth in IoT devices, network infrastructure, and information flow opens up the requirement of security and privacy against a vast category of attacks. Reports (Sobers, 2022; Manship, 2021; Fox, 2021) show that most organizations are unaware even of any type of attack. Post the COVID19 pandemic, most of the organizations have been struck by one or the other types of attacks. The surge in cyber-attacks during this period is due to the lack of infrastructure for cyber defense mechanisms and the negligence of organizations towards security (Nabe, 2021; Schwartz, 2021). These attacks broadly consist of known attacks such as DoS, DDoS, backdoor, data theft as well as unknown attacks inspired by the bugs present in the software.

An Intrusion Detection System (IDS) is a defensive technique that can protect against different types of attacks. This technique can be classified as either signature-based or anomaly-based detection or even as a combination of both. In the former category, attacks can be identified by using attack patterns in traffic whereas, in the latter, they can be detected by learning the historical attack and normal traffic profile. Any deviation from the normal profile is reported as an anomaly. Many attacks remain unnoticed by the network administrators. Such attacks are untraceable due to the limited number of signatures available for the defense mechanisms which are applied to understand the attack patterns. The amount of privacy and financial losses during the pandemic is sufficient in itself to attract the focus of researchers towards Cyber-Security (Lallie et al., 2021; Okereafor, 2021; Gabriel et al., 2021). Several Machine Learning (ML) and Deep Learning (DL)

[*] Corresponding author at: Department of Computer Science & Engineering, Siksha O Anusandhan Deemed to be University, Bhubaneswar, India.
E-mail address: Vika96snz@gmail.com (V. Kumar).

https://doi.org/10.1016/j.cose.2022.103054
0167-4048/© 2022 Elsevier Ltd. All rights reserved.

techniques are already designed to defend against these diverse sets of network attacks. The shortage in the volume of attack footprints not only increases its incompetency in detecting such attacks but also leads to a biasness towards the attacks with higher samples of attack traffics. This results in the inefficiency of the proposed solutions against the minority-attack classes. The proposed work is aimed at handling the issue of data imbalance where some classes possess lower number of samples compared with others. Mozo et al. (2022) have demonstrated the effectiveness of generated attack data. They experimentally demonstrated that the generated data are as effective as the real one without compromising with the privacy of actual data. However, generating traffic samples need some amount of prior distribution to produce statistically significant sample distribution. So, where there is no sample or a negligible number of samples is available, the synthetic data may not be statistically significant for the experimental work. The attack variations are implicitly covered through the diverse sets of synthetic data produced upon the existing samples. The proposed work also generates real samples from the few existing samples in minor classes. The performance of the model infers the quality of these real and trustworthy dataset, discussed in Section 4.

Generative modeling is a new trend attracting the focus of researchers to fill the aforesaid gap (Kim et al., 2020; Andresini et al., 2021; Lee and Park, 2021; Ali-Gombe and Elyan, 2019) and successfully compliments the modeling of cyber defense mechanisms. This technique consists of two deep networks called generative (G) and adversary (D) networks. The first one is responsible for generating synthetic data from a random noise distribution over some pre-specified dimensions. It consists of a discriminator or a critic deep network, which actually acts as the mentor for shaping the generator to learn the realistic pattern of minority classes. And the second one, the adversary (D) network estimates the probability of any data sample belonging to actual training data or the generated data by the generator. This probability is fed back to the generator iteratively until the loss stops changing or has negligible improvement over further iterations. Once the generator is fully trained, it can generate a data sample that follows a distribution similar to the actual training data. Thus, this process generates data points that are not present in actual training and hence, can manifest the unseen attacks and assist in detecting them. There are other approaches to data samples - under-sampling (Hasanin and Khoshgoftaar, 2018) and oversampling (Chawla et al., 2002). The under-sampling approach randomly discards the data samples whereas the over-sampling approach randomly duplicates the samples. In comparison to the aforementioned approaches, the Synthetic Minority Oversampling Technique (SMOTE) (Chawla et al., 2002) performs better. These methods are based on the individual distribution of classes and overlook the distribution of other classes and their effect on data generation. Experimental analysis in Choi et al. (2019) shows a strong sign of exploring other alternatives for data balancing. The papers (Divekar et al., 2018; Zhu et al., 2017) also conclude with the requirement for alternative techniques to tackle with the data imbalance problem for minority classes.

To overcome the aforesaid problem, this paper proposes an XGBoost stacked GAN approach using Auto-Encoder (AE) as a feature extraction to enhance the performance of intrusion detection with an optimal feature vector. At first, this approach implements a Deep Auto-Encoder (DAE) to derive the subset of the most significant features. After that, different GAN models are trained to synthesize the minority attack data samples. The proposed work evaluates these architectures on several benchmark datasets. To demonstrate the robustness of GAN as data synthetization method, this paper applies different classifiers with the proposed framework and it is found that XGBoost classifier gives the highest performance. Further, it is also observed that WCGAN outperforms the other GAN models. The main contributions of the proposed framework are as follows:

1. This paper proposes a data generation model for designing an intrusion detection framework by stacking AE with Wasserstein Conditional GAN (WCGAN) (Zheng et al., 2020) and XGBoost classifier (Chen and Guestrin, 2016) which can detect vast categories of Cyber-Attacks.
2. The framework is first evaluated with a classical training datasets (NSL-KDD (Tavallaee et al., 2009), UNSW-NB15 (Moustafa and Slay, 2015) and BoT-IoT (Koroniotis et al., 2019)) and then, by mixing the actual training with generated data. It is observed that the performance of the classifier is higher for mixed dataset as compared to the original training data.
3. This data generation model is compared with DGM (Dlamini and Fahim, 2021) technique which uses the CGAN model for its framework in contrast with WCGAN with gradient penalty. The proposed framework shows improved performance over that of DGM.
4. The analysis shows that improvement in the performance of attack detection by applying the proposed framework is highly significant and promising which concludes that this work efficiently tackles with the data imbalance problem for minority attack classes.

The rest of the paper is organized as follows. Section 2 reviews some existing work related to this paper. Section 3 presents the proposed methodology. It is followed by the experiments and discussions in Section 4. Finally, Section 5 concludes the paper.

2. Literature review

In this section, several recent works using generative networks dealing with the aspects of cyber-attacks are discussed in brief. Most of the previous works on IDS and Cyber-attack detection techniques did not consider the problem of data imbalance and were biased to certain majority classes. The authors of the papers (Vinayakumar et al., 2019; Manzoor and Kumar, 2017; Zhou et al., 2020; Li et al., 2020; Diro and Chilamkurti, 2018) mainly focused on the detection rate of proposed models. But, in the case of an imbalanced-class dataset, a higher detection rate or accuracy does not truly reflect the performance of the models. It means, a highly accurate model may show lower performance against a class consisting of a very few samples (Gupta et al., 2022; Al and Dener, 2021; Ding et al., 2022). So, this type of data imbalance problem makes the classification model biased towards the classes with higher attack samples. To solve this problem of data imbalance, generative networks are applied to generate synthetic data samples for minor classes that help the classification model to get a good generalization power with lower biasness. These generative models are applied as classifiers in some works by training the discriminator in different ways. The following sections depict the detailed analysis of these types of works.

Huang and Lei (2020) have proposed an imbalanced GAN-based IDS for ad-hoc networks that uses Deep Neural Networks (DNN) as classifiers. Feed-Forward Neural Network (FFNN) is used to generate the feature vector from the raw network traffic and Imbalanced Generative Adversarial Networks (IGAN) is used to generate new samples for the minority classes. This model does not show significant improvement using FFNN as a feature vector generator and DNN as a classifier. On the other hand, Farajzadeh-Zanjani et al. (2021) have used FFNN to propose two Dimensionality Reduction (DR.) techniques using generative adversarial concepts. The first approach is supervised while the second one is unsupervised using the same adversarial concept. These techniques use two FFNNs in the context of an adversary and also apply two constraints such as “affinity correlation” and “separability” to the NN. These constraints result in lower dimension space and act as an objective function to the generator. The paper focused on dimensionality reduction without analyzing the performance of those features during model designing that aims at solving the data imbalance problem using generative modeling.

Li et al. (2019) propose a technique against False Data Injection (FDI) attacks for recovery from the tampering of measurement of Cyber-Physical Systems (CPSs). They have integrated the physical model that captures ideal measurements with a generative model. The generative model captures the deflection from the ideal measure and generates non-tampered data to restore the actual measurement. They have applied an online adaptive window idea to accelerate the training of GAN. Zhang et al. (2020) propose a semi-supervised technique to detect FDI attacks by integrating autoencoder and GAN for smart grid systems. Firstly, the AE is trained for minimizing the reconstruction loss for supervised and unsupervised inputs. The trained AE is then used for the training of GAN. de Araujo-Filho et al. (2020) propose an IDS in a fog environment for cyber-physical systems using GAN and autoencoder. The decoder of the AE is the trained generator of the GAN module. The detection of intrusion is performed by applying an attack detection score proposed in the work. This score is the combination of discriminator loss and generator loss. Siniosoglou et al. (2021) propose an IDS focusing on smart grid systems (SGS) using a novel architecture Autoencoder-Generative adversarial network for detection of anomalies in operational data in SGSs, classification of TCP and DNP3 attacks. These works mainly focus on CPSs and FDI attacks without considering other categories of attacks that have lower footprints.

Chandy et al. (2018) propose an anomaly-based detection model against simulated cyber-attacks using generative models with variational inference. This work explicitly applies to water distribution systems in which hacking of programmable logic controllers, activation of actuators and deception attacks using Variational Auto-Encoder (VAE)are the main focus. The scope of the paper is limited to only two-class problems that work on anomaly-detection. Kim et al. (2018) propose a zero-day malware detection technique that uses the concept of a transferred learning approach. They have applied VAE to learn the malware characteristics by extracting appropriate features. The features are then passed to the GAN model for a stable training. The power of the discriminator is passed down to the detection module. Yang et al. (2019) propose an IDS by combining the conditional VAE and DNN in which the decoder of VAE is responsible for generating attack data samples for minor classes. The weights of hidden layers of DNN are initialized by using the encoder which is used as a classifier in the proposed IDS. But this model poses a higher False Positive Rate (FPR).

Yan et al. (2019) have shown a different angle of using GANs to evade the detection systems. They have used WGAN with gradient penalty that generates synthetic DoS attack data which can be misclassified as normal traffic by any detection module. They have evaluated their work on an NIDS implementing CNN that resulted in a higher drop (47.6% from 97.3%) in detection rate. The scope of the work is limited to generate data samples of DoS attack only that follows the distribution which mimics normal traffic. Li and Li (2020) proposes an evasion and defense technique using a mixture of attacks and a deep ensemble-based generative adversarial approach that helps generate different types of attacks. The generation technique is based on the manipulation of different malwares to derive perturbed malware examples. The scope of the model is limited to malware attacks only.

Nie et al. (2021) have proposed an intrusion detection mechanism using GAN for edge networks against different attacks such as DoS, packet sniffing, and unauthorized access. The proposed work involves three phases. In the first phase, it implements feature selection applying ensemble-based multi-filter feature selection technique (Shawahna et al., 2018), whereas in the second and the third phases, detection of single and multiple attacks is covered using deep GAN architecture. The paper has combined multiple individual trained discriminators on different attacks for the detection of multiple attacks. Liu and Yin (2021) propose a Long Short-Term Memory (LSTM) and a conditional GAN-based technique for synthetic data generation of the low-rate DDoS attacks. CGAN uses the relationships learned by LSTM among the sequenced network packages. Dlamini and Fahim (2021) also present a data generation technique to improve the class imbalance for minority classes in anomaly detection. Generators and discriminators are implemented as FFNNs. Moti et al. (2021) propose a generative approach for generation and detection of novel malware traffic for Internet of Things (IoT) networks. Extraction of high-level data features are achieved through Convolutional Neural Network (CNN) and to capture the dependency among those features are discovered using CNN and LSTM. Ring et al. (2019) propose a GAN-based network flow traffic generation that not only generates continuous features but also generates the features like IP addresses and port numbers. The authors propose to convert these features to continuous numeric transformation, binary transformation and embedding transformation. This paper outlines the advantages and disadvantages of aforesaid transformation methods to generate features. Vuttipittayamongkol and Elyan (2020) have proposed a method to tackle with the problem of data imbalance in medical datasets. Their method is based on under-sampling which exposes the visibility of minor class samples. However, this method may not be effective where the distribution is highly skewed. Also, an imbalanced dataset with fewer or no overlapping in the majority class samples may fail to handle an imbalance class problem. The present paper is an attempt to address the abovementioned issues by applying generative model as an oversampling technique.

Elyan et al. (2021) present a method by integrating the concepts of class decomposition and SMOTE technique to tackle with the data imbalance problem. Though their method has improved over the earlier similar approaches, the k-mean clustering is unreliable in generating a similar class-decomposition as achieved earlier at some other point. Along with this, this approach may also reduce the generalization capability of the trained model on the majority class instance. Also, they have applied SMOTE on minority class samples to increase the number of samples. Against this method, in this paper, we apply the generative model to minor classes only without altering the majority class samples. This method also guarantees to produce minor samples that follow the same distribution as the actual ones.

Ring et al. (2019) have briefly surveyed different intrusion detection datasets. These datasets are classified and analyzed applying different parameters, which may be helpful for research in cyber-attack detection methods. Khraisat et al. (2019) have surveyed the current IDS techniques, datasets used, and the challenges that arise when building a detection model. They also considered the importance of feature selection in IDS design, evaluation metrics, and key challenges in evasion methods. Zheng et al. (2020) have proposed a conditional WGAN with a gradient-penalty-based oversampling technique. They have evaluated the performance of different datasets for binary classification. Engelmann and Lessmann (2021) have also proposed a similar approach for credit scoring context which is also a binary classification. In contrast, in the present paper, we have applied the same generative model for multi-class classification. Also, a symmetric AE is used to deal with the higher dimension and handle the class imbalance problem. In addition, different intrusion detection datasets are evaluated to analyze the performance of the proposed model. Yu et al. (2019) have also proposed similar model but applied the convolutional and de-convolutional networks in building GAN network. They have evaluated the model in time-domain signals for fault data generation for bearings.

de Carvalho Bertoli et al. (2021) proposes a methodology to generate the latest Network Intrusion Detection System (NIDS) dataset using the cloud platform. They take help of kali Linux in producing the latest attack categories by creating a globally distributed virtual testbed. In a similar approach, Ferriyan et al. (2021) have proposed a technique that generates network intrusion dataset using synthetic attack samples. In their work, they firstly generate encrypted network traffic and secondly, create new IDS dataset with encrypted trace. This technique requires physical network setup and choice of attack classes to be performed. In contrary to this, our model does not require any physical setup for generating attack and benign traffic and other preprocessing steps in subsequent stages. Our model also shows promising results while evaluating performance. Cordero et al. (2021) have proposed an open-source toolkit that generates attack data in *pcap* format. This tool needs an input traffic along with parameter from the user to specify the attack types and produces *pcap* file as labelled attack dataset. This tool is limited to IP4 data traffic and also works for attacks that do not alter the network state. It also assumes that the input traffic is free from any attack traffic.

In state-of-the-art, it is found that GAN is mainly applied as a feature generator and classifier where the discriminator of trained GAN is applied as a classifier. In many papers, GAN is applied only for anomaly detection and dimensionality reduction. Very few works are focused on the issues of the data imbalance problem. The aforesaid works also lack the exploration of WCGAN with gradient penalty as a solution to the data imbalance problem and using it for proposing a multi-class classification model.

3. Methodology

In this paper, generative models, particularly Generative Adversarial Networks (GANs) are extensively explored for data generation. The proposed framework is designed using Wasserstein Conditional GAN (WCGAN) (Zheng et al., 2020) with XGBoost classifier (Chen and Guestrin, 2016). The proposed model shows a significant footprinting of attacks to tackle with the data imbalance during the design of the detection model. Basic theoretical descriptions of GAN variants used in this work are explained in the following subsections.

3.1. Generative adversarial networks

As it is discussed earlier in this paper, the generative adversarial networks are a two-player game in which neural networks update their weights to have better data generation and discrimination capability. Fig. 1 shows the general view of standard GAN and CGAN with two hidden layers where –

- z denotes the random noise input and $z|y$denotes the generator's input conditioned on the labelyrespectively.
- x and $x|y$ denote the actual training data and actual training data conditioned on label$y$.
- x^* denotes the generated data from standard GAN whereas, $x^*|y$ is the data generated by the conditional GAN conditioned on label$y$.

The weight update process is achieved through back-propagation where, at the time of updating the generator, the discriminator should be kept fixed. The Wasserstein version and the standard GAN differ from each other only in terms of loss calculation and weight updation.

Standard GAN: This is the simplest model among other GANs in which the objective of discriminator (D) is to predict the probability of a given data belonging to a real or fake distribution where the generator (G) is responsible for generating fake data samples. Suppose, Z is a random noise distribution which forms a base for generating diverse synthetic output mimicking the distribution of actual data samples.

The generator's efficiency depends on this learned distribution by the fact how close it is to the actual one. The whole model can be seen as a two-player game where the training process is accomplished by G and D working in opposition with each other. Let's say–

- P_G is the distribution of generated data over the actual data X;
- P_X is the distribution of actual data; and
- P_Z is the distribution of generated data from the noisy input.

Now, the objective of this game is defined by Eq. (1) in which the discriminator tries to maximize the expected value $E_{X\sim P_X}[D(X))]$ and the generator indirectly minimizes the expected value $E_{Z\sim P_Z(Z)}[D(G(Z)))]$. Here, $D(X)$ is the probability estimation of D that a real data sample X is real and $G(Z)$ is the output of G when noise Z is provided as input. $D(G(Z))$ is the probability estimation of D predicting generated data as real? Now, the minimax equation is as follows:

$$minmax\ \mathcal{L}(G,D) = \ E_{X\sim P_X(X)}[D(X))] + \ E_{Z\sim P_Z(Z)}[D(G(Z)))] \tag{1}$$

The problem with this method is vanishing gradient when the discriminator is more accurate i.e., $D(x) = 1\ \forall x\epsilon P_X$ and $D(x) = 0\ \forall x\epsilon P_G$ which results in $\mathcal{L}(G,D) = 0$. Hence, no gradient update takes place at the time of learning as there is no loss. Another major issue with this approach is the mode collapse where the generator is stuck at a stage producing the same data. It leads to the poor learning of realism of synthetic data generated by Gas compared to the actual data distribution. Since the loss function used by this GAN computes the JS divergence of two probability distributions P_X and P_G, it fails when both distributions are disjoint to each other.

WGAN: In place of JS divergence another metric called *Wasserstein distance* (or, Earth Mover's distance) is proposed that addresses the drawbacks of the earlier metric. This metric is the measure of distance between probability distributions P_X and P_G. This distance signifies the amount or value that is required to convert one probability distribution to the shape of another and is mathematically given by the following equation.

$$W(P_X, P_G) = \frac{1}{K}\,{}^{sup}_{\|f\|_L \le K} E_{X\sim P_X}[f(x)] - E_{X\sim P_G}[f(x)] \tag{2}$$

Here, function f is the Wasserstein metric which should be a K-Lipschitz function i.e., it satisfies the condition $||f||_L \le K$ and K is a Lipschitz constant. In this GAN model, the discriminator compared to the previous model does not work on telling how apart the generated or fake data sample is from the actual one. Rather, it learns the K-Lipschitz function that helps in computing the Wasserstein distance. Eq. (3) shows the mathematical definition of the distance as a loss function between P_X and P_G. As the loss gets decreasing in the training phase, this distance becomes smaller and leads to the generator's output closer to the actual distribution.

$$\begin{aligned}\mathcal{L}(P_X, P_G) = W(P_X, P_G) = w \in W max E_{X\sim P_X}[f_w(x)] \\ - E_{X\sim P_G(Z)}[f_w(g_\theta(Z)]\end{aligned} \tag{3}$$

Where, f_w is parameterized by $w \in W$.

The objective of the WGAN is to learn the parameter w to get a good estimation of the function f_w using Eq. (3) which computes the Wasserstein distance between P_X and P_G. In paper (Gulrajani et al., 2017) authors mentioned the problem of maintaining K-Lipschitz continuity of the function f_w at the time of training. To address this problem, they have introduced the clamping of weights w after every gradient update to a smaller range

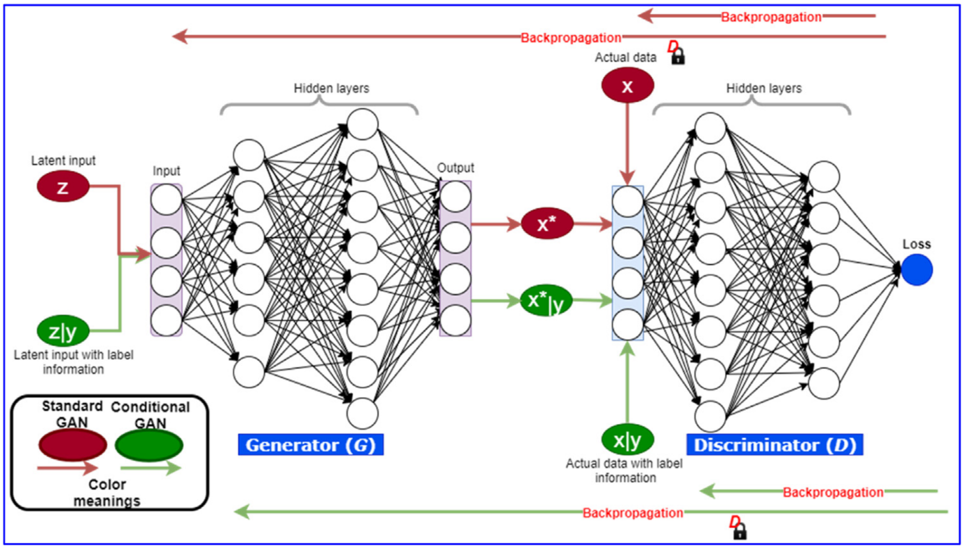

Fig. 1. Architecture of standard GAN and CGAN.

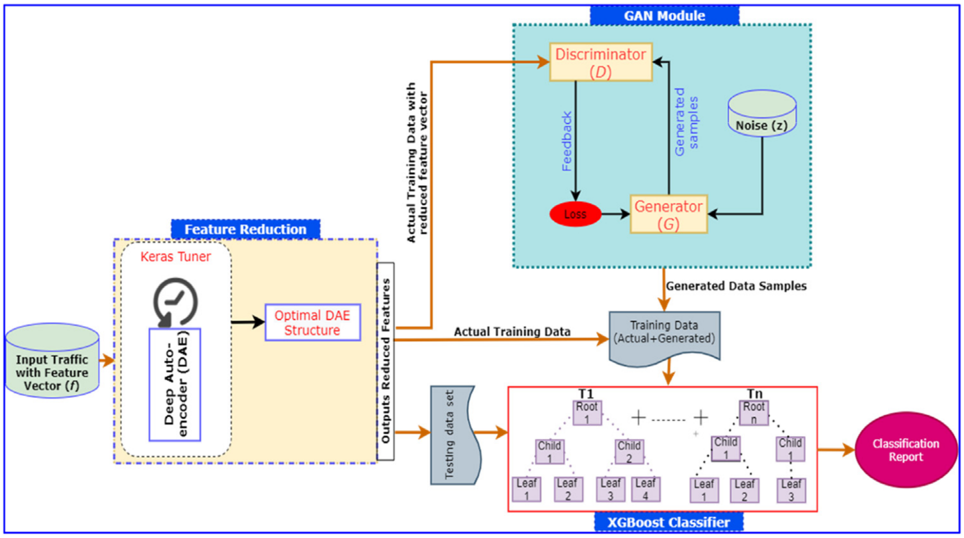

Fig. 2. Proposed architecture of the framework.

resulting in compaction of parameter space W. This helps f_w to preserve the continuity by obtaining its lower and upper limits which results in stable learning of generative models. Four different types of GAN models are analyzed in this paper which are standard GAN or GAN, Conditional GAN (*CGAN*), WGAN, *WCGAN*, and among them, *WCGAN* gives significant results compared to other GAN models. In our proposed model WCGAN is applied during the attack data generation process.

3.2. Proposed model

Fig. 2 illustrates the proposed framework as a minority class data generation and detection of those attacks. First AE is applied to get a reduced feature vector and the configuration is chosen by selecting the hyper-parameters using keras tuner library. After selecting the optimal structure of the AE, the dataset is fed to get the reduced feature vectors. To demonstrate the optimality of the reduced feature vector, a comparison of classifiers is performed before and after feature reduction. Next, the GAN model's generator is provided with a randomly initialized noise distribution (z) and based on these inputs, it generates data samples for minority classes. The generated samples are fed to the discriminator along with the actual training data. GAN is a two-player game where the discriminator tries to maximize the correct prediction for generated and actual training attack data samples by producing minimum error or loss. On the other hand, the generator tries to maximize that error by generating more and more realistic data samples that match the actual training attack data samples.

Iteratively, the generator is trained by the discriminator's loss feedback to produce the synthetic data that covers minority attack classes. These synthetic data are then used to overcome the imbalanced nature of the dataset, reduce the biased learning of any model and maximize the generalization capacity of any model. After training the individual GANs one by one, they are evaluated based on loss value and compared with each other (refer to Section 4).

The model that performs better among them is taken as the final one. WCGAN gives better results and the data samples of WCGAN are considered for training the classifier (discussed in Section 4). We have trained our models on different classifiers (i.e.,

Table 1
Hyper-parameters setting.

GAN		XGBoost	
Hyper-parameters	Standard GAN/WGAN	Hyper-parameters	Value under assumption
No. of hidden layers	6	Objective	Multi:softmax
Activation	Relu/sigmoid	nfolds	3/5
Learning rate	5e-4, 1e-4	Learning rate	0.1
Batch_size	200	Colsample_bytree	0.5
Log interval	100	Max_depth	8
	100	Metrics	mlogloss
Critic_pre_train_steps			

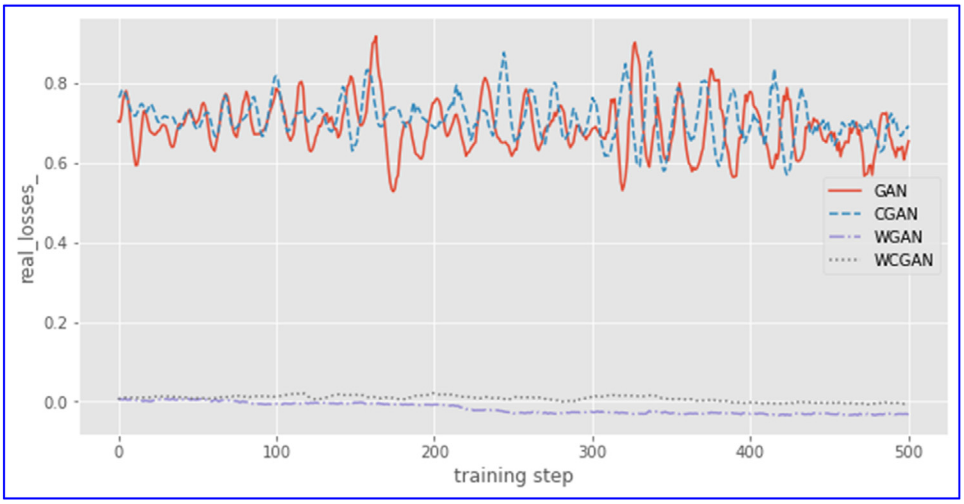

Fig. 3. Discriminator loss on real training data on UNSW-NB15.

XGBoost, Random Forest, Decision Tree, Support Vector Machine) and among them, XGBoost gives higher performance compared to others.

The XGBoost classifier is evaluated to illustrate the significance of GAN for overcoming the data imbalance problem. The data samples generated by the trained generator are mixed with the actual training data. This mixed training data is then used to train the classifier model. The settings of the GAN architectures and XGBoost are given in Table 1. GAN setting is applied to all the variants used in the analysis (i.e., Standard GAN, CGAN, WGAN and WCGAN). *Critic_pre_train_steps* parameter is particular to WGAN architecture, in which the term critic is used in place of the discriminator. This critic network does not involve in the direct decision about actual or generated samples like the discriminator network. Referring to Figs. 3 and 4, we observe that the performance of both WGAN and WCGAN show lower loss compared to other GAN models. Now, since this paper works on supervised data generation, among the two variants of the Wasserstein model, WCGAN is chosen for the implementation of the proposed model.

Since the type of problem is multi-class classification, the XGBoost uses the objective function “*multi-softmax*” for conditional variants whereas for other variants, it uses the *binary-regression* objective function. Details of the structure of GAN model is given in Table 2 in more general terms that are applied to all the dataset considered in this paper. Here, the notation n denotes the random input neurons decided for 1st hidden layer and f is the actual feature vector space without label column considered in the proposed work. In case of conditional variants of the GAN, the discriminator gets the data generated by the generator G along with the label column appended with it.

Table 2
General structure of the variants of GAN for all dataset.

		Generator		Discriminator/Critic	
#	Layer	Size	Activation	Size	Activation
1	Fully connected	n	Relu	2*n	Relu
2	Fully connected	2*n	Relu	n	Relu
3	Fully connected	$f/(f+1)$	Relu	1	Sigmoid

4. Experiment and discussion

This section of the paper discusses the experimental setup and the dataset along with the performance analysis of the proposed model.

4.1. Benchmark dataset

The proposed framework is analyzed on three different datasets. The details of each dataset are provided in this subsection.

NSL-KDD: This is an improved version of the original KDDcup99 developed by Tavallaee et al. (2009). This dataset endures some drawbacks mentioned by McHugh (2000) and this cannot be used as an exact representation of existing real traffic. The full dataset as training and testing is available under the file KDDTrain+ and KDDTest+. It consists of 43 features with a total of 23 classes in the training file and 38 in the test file. We have removed some

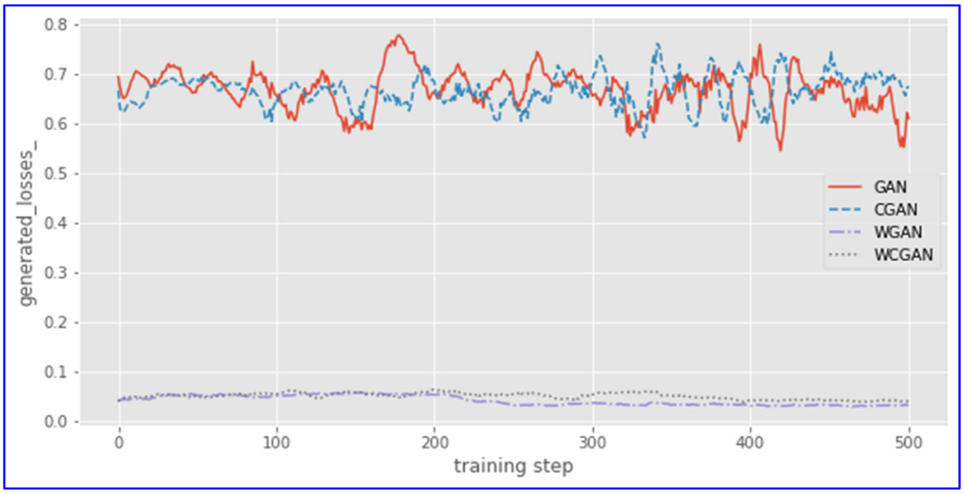

Fig. 4. Discriminator loss on generated data.

Table 3
Dataset distribution under different categories.

NSL-KDD		UNSW-NB15		BoT-IoT	
Normal	67,343	Normal	56,000	Normal	370
DoS	45,927	DoS	12,264	DoS	1,320,148
Probe	11,656	Reconnaissance	10,491	Reconnaissance	72,919
R2L	995	Fuzzers	18,184	DDoS	1,541,315
U2R	52	Backdoor	1746	Theft	65
		Generic	40,000		
		Analysis	2000		
		Exploits	33,393		
		Shellcode	1133		
		Worms	130		

features such as “num_outbound_cmds” that have all the cells as 0, “outcome” and “difficulty”. Also, for the classification purpose, we combined the diverse attack classes into a more abstract number of classes shown in Table 3.

UNSW-NB15: This dataset is generated at the cyber range lab of the Australian center for Cyber Security (ACCS) using the IXIA PerfectStorm tool. It consists of modern attack patterns in traffic along with genuine network traffic. The dataset has 9 attack classes and one normal class (refer to Table 3) with a total of 47 features. These features are extracted using Argus and Bro-IDS tools. Moustafa et al. presented the dataset in Moustafa and Slay (2015) where they claimed that the dataset overcomes several existing issues found in previous datasets, e.g., the lack of current network traffic patterns and attack footprints.

BoT-IoT: This dataset consists of legitimate and other attacks of IoT network traffic (Koroniotis et al., 2019). Koroniotis et al. (2019) evaluated the trustworthiness of the dataset using statistical and ML techniques and compared it with other benchmark datasets. The purpose of the dataset is to overcome the issues of existing datasets like the reliability of labeled data, lack of botnet attack patterns, redundancy. Table 3 shows the details of each dataset discussed above.

4.2. Data distribution post applying GAN model

The objective of the synthetic data generation is to have plenty of data samples in each of the minor classes to train any classification model perfectly. Table 3 above shows the distribution of different categories in each dataset prior to applying the generative step. We have considered only classes with low number of samples for data generation. The proposed model applies GAN to generate the synthetic samples of different minority classes in order to get the higher number of representative samples for training a classification model. This helps the model to enhance its generalization capability on unseen data samples. The distribution of classes in training set after combining the synthetic samples is shown in Table 4. These data samples are explicitly used to model XGBoost where we have applied k-fold cross validation with the value of k as 3, 5 and 5 for NSL-KDD, UNSW-NB15, and BoT-IoT respectively. The majority class samples will remain unchanged as the mixing of synthetic data is limited to minority classes. The test samples under consideration consist of actual test data samples of each one available in the respective repository. The test-set consists of 22,542 samples of NSL-KDD, 82,332 of UNSW-NB15, and 733,705 of BoT-IoT which are approximately 15%, 20%, and 24% of the training data (given in Table 4) respectively.

4.3. Evaluation metrics

The performance of the proposed framework is analyzed using the following metrics.

Precision: It is the rate at which a particular class is predicted correctly without the misclassification of other classes into this class. In other words, it is the accuracy of a class that reflects that the other classes present in the dataset are not misclassified to this particular class. Mathematically it is given by the following equation.

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

Recall: In contrast to precision, it refers to the accuracy with which the classified samples of a particular class belong to itself

Table 4
Data distribution under different categories after applying generator.

NSL-KDD		UNSW-NB15		BoT-IoT	
Class	Sample size	Class	Sample size	Class	Sample size
Normal	67,343	Normal	56,000	Normal	72,919
DoS	45,927	DoS	40,000	DoS	1,320,148
Probe	11,656	Reconnaissance	40,000	Reconnaissance	72,919
R2L	11,656	Fuzzers	40,000	DDoS	1,541,315
U2R	11,656	Backdoor	40,000	Theft	72,919
		Generic	40,000		
		Analysis	40,000		
		Exploits	40,000		
		Shellcode	40,000		
		Worms	40,000		
Total	**148,238**		**416,000**		**3,080,220**

rather than misclassified in other classes. So, we can say, in precision, the focus is on the prediction of samples to a particular predicted class, but in recall the focus shifts to a particular actual attack class whose instances are predicted to different classes. It is calculated using the following equation.

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

F1 score: It is the measure of the balance between precision and recall. The higher the score, the better is the performance of precision and recall. It can be mathematically given in the following equation.

$$F_1 = 2 * \frac{precision * recall}{precision + recall} \tag{6}$$

False Alarm Rate (FAR): It is the measure of how frequently a genuine data sample is misclassified by the model. The mathematical expression for this is given in the following equation.

$$FAR = \frac{Correctclassifiedinstancesofgenuinedata}{Totalnumberofinstancesofgenuinedata} \tag{7}$$

4.4. Performance analysis

This section is subdivided into two sub-sections. The first subsection discusses the performance of different GAN trainings while in the second one, the performance of the proposed model is depicted.

4.4.1. Comparison of different variants of GAN models

In this section, the performance of different GAN models is analyzed and compared to visualize the significant model for the proposed work. The standard GAN and WGAN are unsupervised approaches to the generative models that are also compared with their conditional variants. Fig. 2 shows the discriminator loss on real training data, in which the standard and the conditional GAN show higher losses as compared to that of WGAN and WCGAN. In Fig. 3, the losses on generated data are plotted for the discriminator. By observing Figs. 3 and 4, it can be concluded that the losses for both the real and the generated data are lower for the Wasserstein variant of GAN in comparison to that of standard GAN and conditional GAN. For this reason, WCGAN - the supervised variant of WGAN - is chosen as per the need for supervised training of GAN.

While drawing a comparison between the Figs. 3 and 4, it is observed that the discriminator/critic loss on generated data has slightly increased against the real loss. This increase in D-loss reflects that the generated samples make the discriminator fail the prediction whether the sample belongs to actual or generated data. Hence, the generated data are true representatives of each class. In this paper, WCGAN is used to propose the framework to address the problem of imbalanced classes. As a result, the ML models can easily generalize the detection of those classes with a higher level of correctness. In Fig. 5, a comparison between WGAN and WCGAN on critic loss is shown. This loss is computed based on the estimated Earth-Mover (EM) distance (Eq. (2)) between the real and the generated data. It is evident again that WCGAN shows a lower loss compared to that of WGAN. From the aforesaid discussions, it is concluded that WCGAN performs better than the others. This is why the proposed framework is designed based on WCGAN.

Table 5
Performance of proposed model on different ML models.

ML Models	Dataset	Precision	Recall	F1 score	FAR
XGBoost	NSL-KDD	96.66	99.65	98.13	1.44
	UNSW-NB15	81.39	81.54	81.46	12.50
	BoT-IoT	99.42	99.16	99.29	1.33
Random Forest	NSL-KDD	91.30	86.82	89.00	6.87
	UNSW-NB15	82.36	70.94	76.22	9.60
	BoT-IoT	93.74	94.13	93.93	14.10
Decision tree	NSL-KDD	79.55	73.40	76.35	7.33
	UNSW-NB15	70.85	71.39	71.12	14.22
	BoT-IoT	86.64	79.22	82.76	4.67
SVM	NSL-KDD	18.55	43.06	25.93	7.34
	UNSW-NB15	72.07	53.83	61.63	14.52
	BoT-IoT	72.82	70.06	71.41	9.81

4.4.2. Performance of the proposed model on original and mixed training data

a) *Analysis of proposed model on different ML models*

After selecting the WCGAN with gradient penalty, different ML models are executed on each dataset by mixing the generated data with the actual training samples. Table 5 shows the performance metrics on all the classifiers obtained for each dataset. The metrics are calculated on all the categories present in the dataset, i.e., normal and attack. From this table, it is evident that the XGBoost model outperforms the other models. As a result, to analyze the proposed model, an XGBoost classifier along with WCGAN is applied.

a) *Analysis of the performance of XGBoost classifier before and after applying the proposed model*

In this part, the proposed model is analyzed in detail. Fig. 6 shows the discriminator loss for each dataset as the training proceeds. Then, a comparison of the performance of XGBoost with and without the proposed model is drawn.

- *NSL-KDD*: This dataset consists of various attack classes which pose a high imbalance among them. To address this imbalance, samples for each class are generated using the framework. The

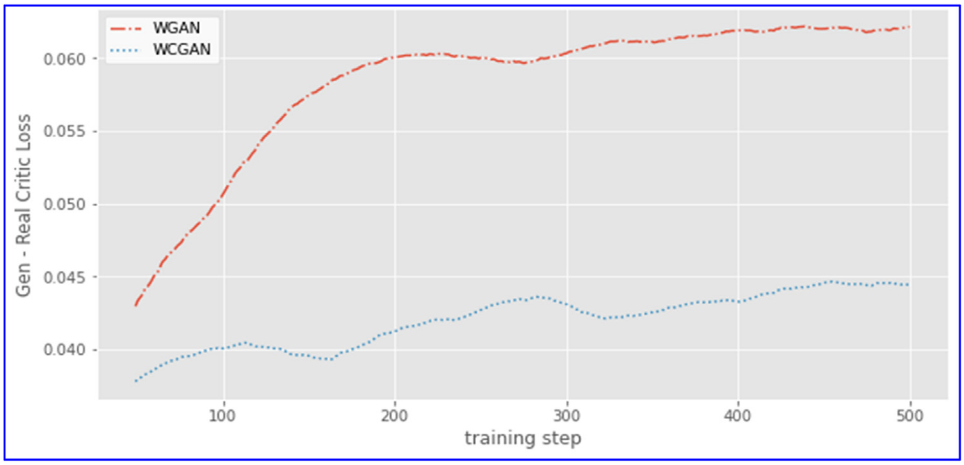

Fig. 5. Difference between critic loss on generated and real samples.

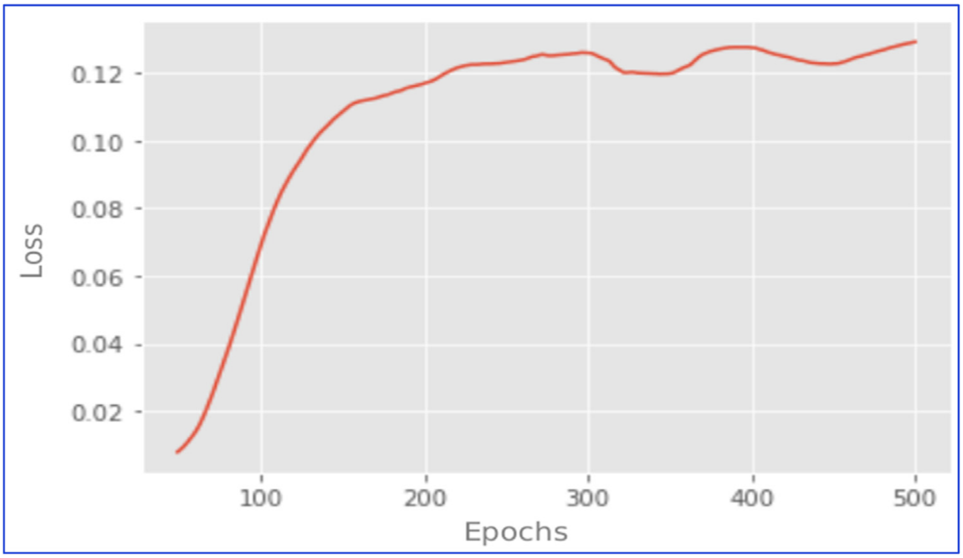

Fig. 6. Mean discriminator loss as the training steps advance on NSL-KDD.

discriminator loss against the training epochs is shown in Fig. 6, where the x-axis denotes the number of training epochs and the y-axis denotes the discriminator loss. From the figure, it can also be seen that the loss is very low in the beginning of the training, whereas a sharp increase of loss (up to 200 training steps) followed by a gradual increase (of up to 500 steps) is observed. A close look at the graph makes it clear that the training steps of 400 is showing approximately the same loss to that of the higher number of steps. For this reason, 400 epochs are selected to train the proposed model for this dataset.

In Fig. 7, the dispersion of generated data is compared with the real training data. A total of 21 classes is generated and plotted against the real data distribution. The generated data shows a similar trend to the real one. This plot shows the particular distribution when “count” and “is_guest_login” features are taken together based on the feature importance. The other combinations also show a similar type of distribution that reflects the realism of the generated data from the proposed framework.

The performance of the proposed framework that uses the XGBoost classifier to evaluate the data generation quality is given in Fig. 8. The figure suggests that the performance of the proposed framework has shown a significant improvement in performance after mixing the generated data with actual training data. At the same time, it also reduces the False Alarm Rate (FAR).

- *UNSW-NB15:* Discriminator loss (i.e., mean loss) on this dataset shows a spike of up to 100 training steps and then fluctuations can be seen (Fig. 9). The x-axis of the plot is the number of training epochs and the y-axis of the plot is the discriminator loss of up to 500 training steps. The amount of loss, in this case, is lower (approx. 0.01–0.10) than that of NSL-KDD which ranges from 0.00 to 0.14 approximately. The stable fluctuation is observed between 0.09 and 0.10. In this case, the trained model is also taken at 400 training steps to generate the synthetic data. Fig. 10 shows the scatter plot of real training and the generated data samples. This dataset consists of 10 categories (refer to Table 3). The data distribution against each category is shown on the left side of the diagram and the generated data samples are plotted on the right side of the diagram to correspond to each category. Plots of data distribution are generated by pairing features that are sorted by their importance using XGBoost. Fig. 10 shows a plot against two features “dmean” and “smean”.

Fig. 11 shows the performance of the proposed framework on actual and mixed training data of UNSW-NB15. The performance of the proposed framework on UNSW-NB15 shows a lower improvement when compared to NSL-KDD. The main reason behind the

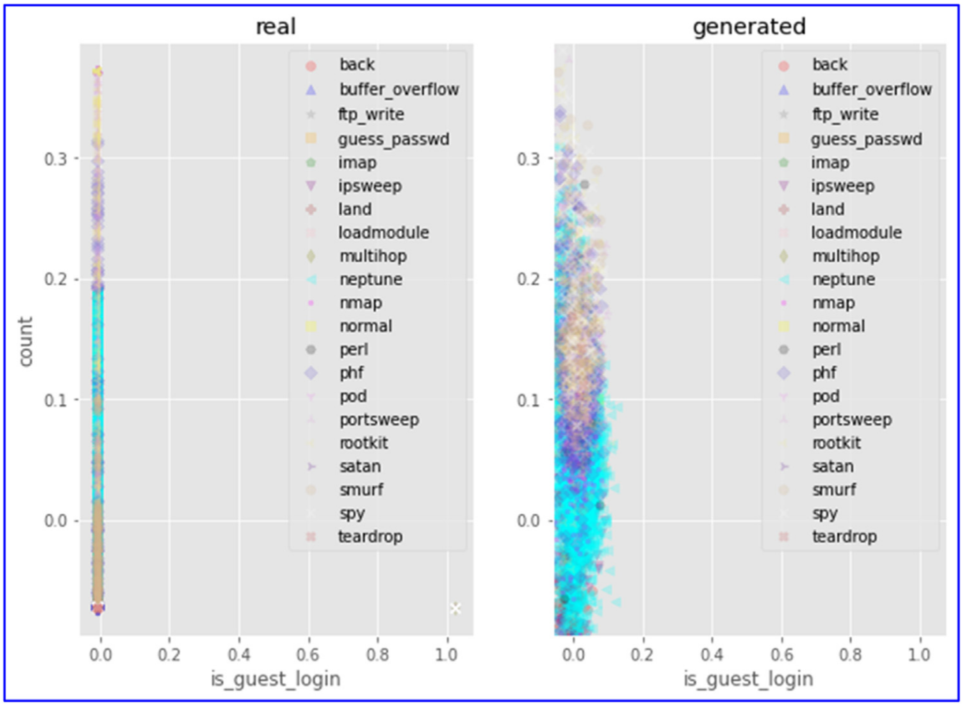

Fig. 7. Real and generated data distribution for the combination of two feature.

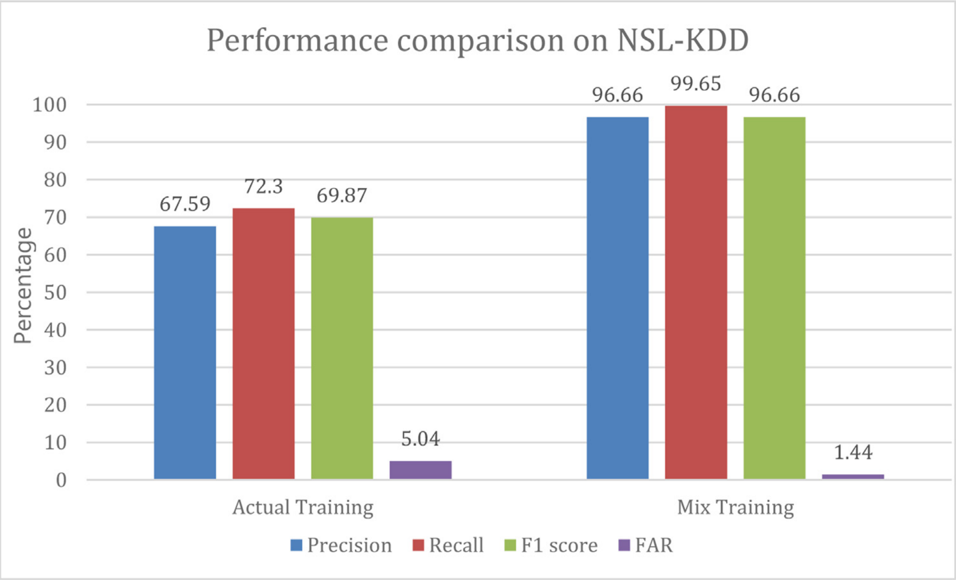

Fig. 8. Performance of mix training data derived from proposed framework against actual training data.

lower improvement is that the actual training data has a lower number of imbalance classes as compared to that of NSL-KDD. In addition, the improvement is contributed by minority classes through the addition of generated data with the actual minority training samples. Hence, the lower number of imbalance classes reflects that the actual data itself contains representative data samples resulting in highly accurate models. This mixing of generated minority class data samples gives higher performance than the performance obtained by applying the actual training data only.

- *BoT-IoT:* On this dataset, the discriminator shows the lowest change in discriminator loss ranging from 0.01 to 0.06 (refer to Fig. 12) with the same training steps and axis labels as the two datasets mentioned above. The maximum spike of up to 200 steps is observed. After that, a gradual fluctuation (roughly 0.06–0.065) of the loss is observed. In this case, a training step of size 500 is selected for the final generator model during data generation. Fig. 13 displays the comparison between the real and the generated data for the combination of two features "seq" and "proto". The distribution is plotted against 5 different categories (i.e., DDoS, DoS, Normal, Reconnaissance and Theft) present in the original dataset.

Now, the performance of the framework on BoT-IoT is shown in Fig. 14. It has better metrics values in both the situations - on actual training and after mixing the generated data compared with NSL-KDD and UNSW-NB15. This higher performance is due to the

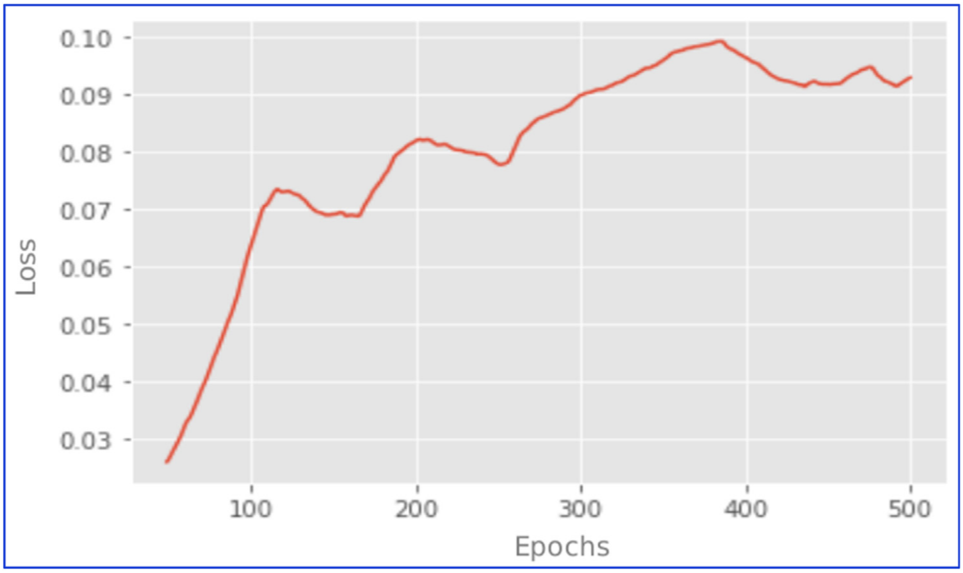

Fig. 9. Mean discriminator loss as the training steps advance on UNSW-NB15.

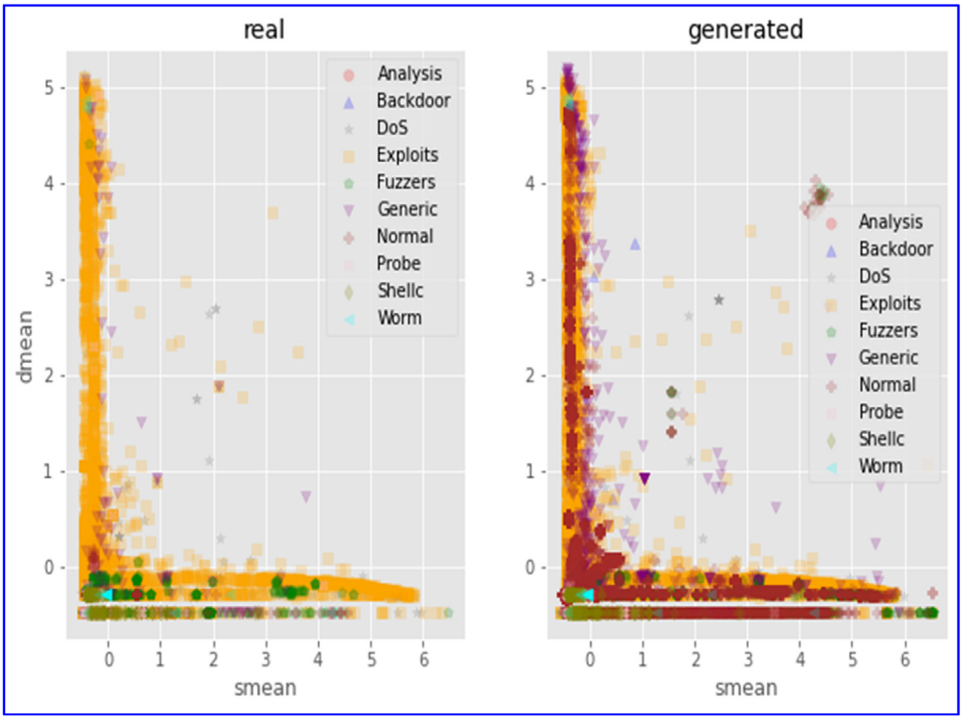

Fig. 10. Real and generated data distribution for the combination of two feature.

presence of lower number of classes in the dataset and sufficient representation of attack traffic of most of these classes.

a) *Comparison of the proposed model with state-of-the-art DGM (Dlamini and Fahim, 2021)*

The proposed model is compared with existing state-of-the-art DGM (Dlamini and Fahim, 2021), in which authors have applied CGAN to propose their models and evaluated it using two datasets - NSL-KDD and UNSW-NB15. DGM (Dlamini and Fahim, 2021) has analyzed performance metrics for each attack class separately. To make it comparable with our proposed model, the individual performance metrics of DGM (Dlamini and Fahim, 2021) attack classes are converted to the overall performance of each model of DGM (Dlamini and Fahim, 2021) by taking the average value of precision, recall and F1 score. As a result, the precision, recall and F1 score (Table 6) corresponding to DGM are the average values calculated for each model. The metrics corresponding to the proposed model are calculated by excluding normal class data samples from the confusion matrix.

After observing Table 6, it is concluded that the proposed WCGAN with gradient penalty has a better performance than that of the CGAN based DGM (Dlamini and Fahim, 2021) model. The rea-

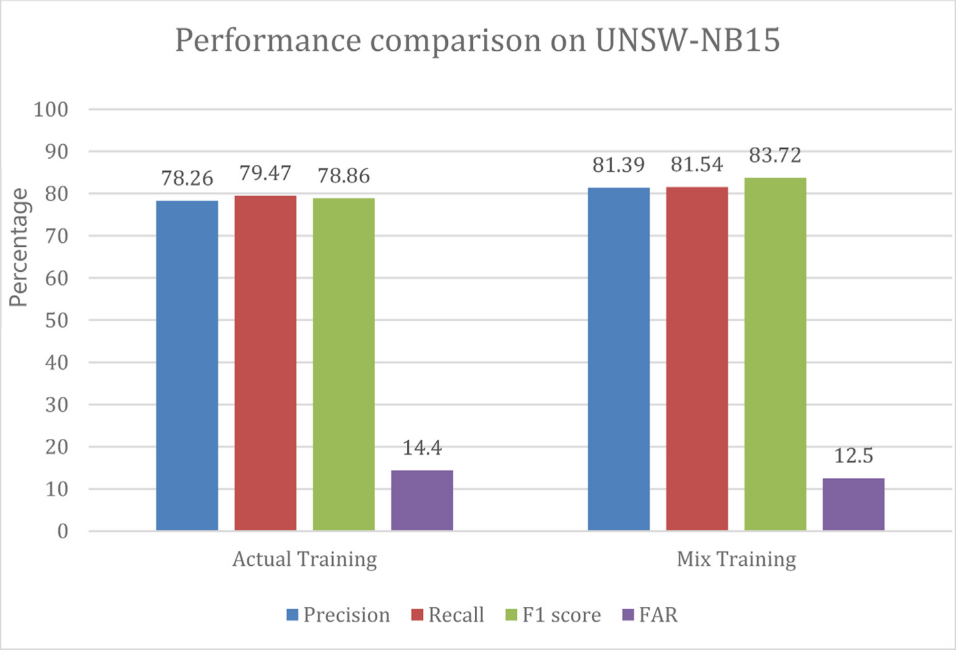

Fig. 11. Performance of proposed framework against actual UNSW-NB15 training data and mix training data.

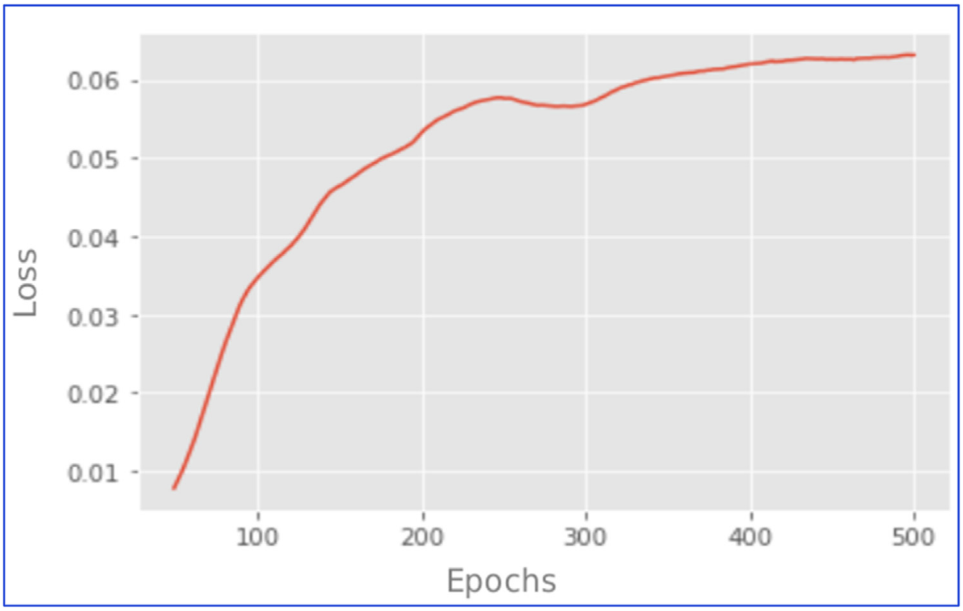

Fig. 12. Mean discriminator loss as the training steps advance on BoT-IoT.

Table 6
Performance comparison of the proposed model with DGM (Dlamini and Fahim, 2021) on attack classes.

ML Model		Proposed Model			DGM		
		Precision	Recall	F1 score	Precision	Recall	F1 score
NSL-KDD	Random forest	83.32	72.30	77.42	63.50	54.50	58.80
	XGBoost	86.70	88.47	87.58	63.82	57.43	60.46
	Decision tree	69.24	68.80	69.02	51.75	53.54	52.63
	SVM	12.71	34.62	18.59	61.00	54.00	57.29
UNSW-NB15	Random forest	67.74	58.61	62.84	45.78	42.11	43.87
	XGBoost	72.80	73.10	72.95	48.38	52.67	50.43
	Decision tree	65.20	66.80	65.99	44.33	43.11	43.71
	SVM	53.27	43.97	48.17	44.22	36.00	32.44

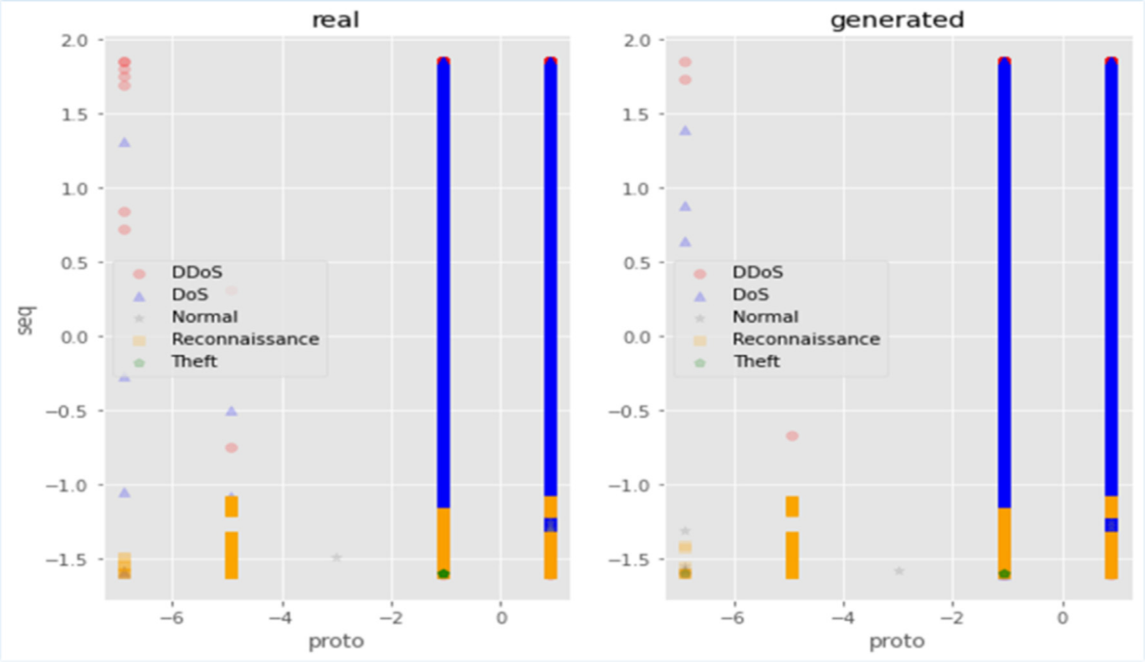

Fig. 13. Real and generated data distribution for the combination of two feature.

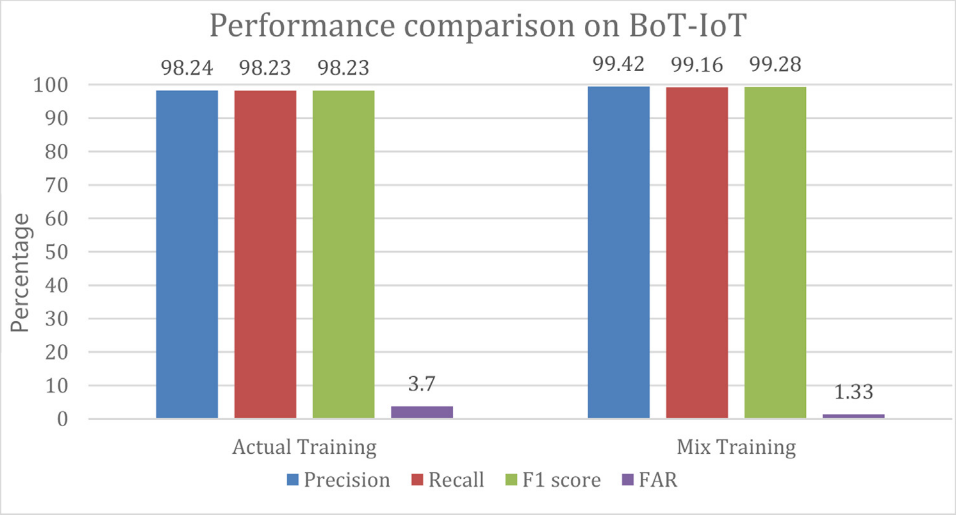

Fig. 14. Performance of proposed framework against actual BoT-IoT training data and mix training data.

son behind this improvement is the objective function of WGAN with gradient penalty that gives better stability to the model and minimizes the loss.

4.4.3. Statistical significance test of proposed model

To signify the performance improvement of the proposed model, this paper has performed a statistical analysis using T-test hypothesis testing. It assumes that the F1 score of the population is less than the one reported in this paper as null hypothesis H_0 and alternate hypothesis H_a assumes that the F1 score is greater than or equal to the reported value. The XGBoost model is executed for 10 times on each dataset. The significance level for p-test is selected as 5% (0.05). The value of p-test at the 0.05 significance level is shown in Table 7. From the table, it is clear that the p-value is much below the significance level and hence, the performance of the proposed model is statistically significant and not occurred by chance.

Table 7
P-value at 0.05 significance level.

Dataset	Proposed model
NSL-KDD	4.22e−5
UNSW-NB15	2.77e−3
BoT-IoT	1.61e−3

5. Conclusion

This paper proposes an XGBoost-WCGAN based data generation model to design an intrusion detection system for multi-class classification that uses gradient penalty for weight update. This data generation model efficiently tackles with the minority attack data generation problems. The proposed model is evaluated on three different datasets NSL-KDD, UNSW-NB15, and BoT-IoT. Different GAN models are first evaluated to observe the learning stability and loss after which WCGAN is selected. The data generated through the WCGAN with gradient penalty for each minor class are then mixed with actual training data samples. Different ML models (XGBoost, Random Forest, Decision Tree and SVM) are applied in our proposed framework. Finally, XGBoost is selected for the proposed model based on the performance. The model is also compared with similar state-of-the-art DGM (Dlamini and Fahim, 2021) which uses CGAN for data generation. Except for SVM, all other ML models for NSL-KDD and UNSW-NB15 datasets are compared with the proposed model. The proposed model outperforms the DGM (Dlamini and Fahim, 2021) model. Based on results obtained, it can be concluded that the proposed model shows a promising performance on all the datasets. And so, it can be helpful to handle difficulties in detecting novel attacks posing very limited number of publicly available samples. Apart from IDS, it can also be applied to different areas such as medical diagnosis system for detecting rare diseases, financial anomalies where the target class samples are either highly imbalanced or samples are very few in numbers, and such others.

In future, rather than using XGBoost or other ML techniques for classification, the possibility of using GAN for both sample generation and classification can be explored to enhance the performance of attack detection. Furthermore, the paper creates ample scope for exploring diverse aspects of evolutionary-based algorithms in enhancing the performance of generative models.

Compliance with ethical standards

Ethical approval: This article does not contain any studies with human participants or animals performed by any of the authors.

Declaration of Competing Interest

The authors declare that they have no conflict of interest.

Data availability

All the data used in this work are cited in the manuscript.

References

Al, S., Dener, M., 2021. STL-HDL: a new hybrid network intrusion detection system for imbalanced dataset on big data environment. Comput. Secur. 110, 102435.

Ali-Gombe, A., Elyan, E., 2019. MFC-GAN: class-imbalanced dataset classification using multiple fake class generative adversarial network. Neurocomputing 361, 212–221.

Andresini, G., Appice, A., De Rose, L., Malerba, D., 2021. GAN augmentation to deal with imbalance in imaging-based intrusion detection. Fut. Gen. Comput. Syst. 123, 108–127.

Chandy, S.E., Rasekh, A., Barker, Z.A., & Shafiee, M.E., 2019. Cyberattack detection using deep generative models with variational inference. J. Water Resour. Plan. Manag., 145(2), 04018093.

Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P., 2002. SMOTE: synthetic minority over-sampling technique. J. Artif. Intell. Res. 16, 321–357.

Chen, T., Guestrin, C., 2016. Xgboost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794.

Choi, H., Kim, M., Lee, G., Kim, W., 2019. Unsupervised learning approach for network intrusion detection system using autoencoders. J. Supercomput. 75 (9), 5597–5621.

Cordero, C.G., Vasilomanolakis, E., Wainakh, A., Mühlhäuser, M., Nadjm-Tehrani, S., 2021. On generating network traffic datasets with synthetic attacks for intrusion detection. *ACM Trans. Privacy Secur. (TOPS)* 24 (2), 1–39.

de Araujo-Filho, P.F., Kaddoum, G., Campelo, D.R., Santos, A.G., Macêdo, D., Zanchettin, C., 2020. Intrusion detection for cyber–physical systems using generative adversarial networks in fog environment. IEEE Internet Things J. 8 (8), 6247–6256.

de Carvalho Bertoli, G., Alves Pereira Junior, L., Alves Neto Verri, F., dos Santos, A.L., & Saotome, O. (2021). Bridging the gap to real-world for network intrusion detection systems with data-centric approach. *arXiv e-prints, arXiv*-2110.

Ding, H., Chen, L., Dong, L., Fu, Z., Cui, X., 2022. Imbalanced data classification: a KNN and generative adversarial networks-based hybrid approach for intrusion detection. Fut. Gen. Comput. Syst. 131, 240–254.

Diro, A.A., Chilamkurti, N., 2018. Distributed attack detection scheme using deep learning approach for Internet of Things. Fut. Gen. Comput. Syst. 82, 761–768.

Divekar, A., Parekh, M., Savla, V., Mishra, R., Shirole, M., 2018. Benchmarking datasets for anomaly-based network intrusion detection: KDD CUP 99 alternatives. In: *2018 IEEE 3rd International Conference on Computing, Communication and Security* (ICCCS). IEEE, pp. 1–8.

Dlamini, G., Fahim, M., 2021. DGM: a data generative model to improve minority class presence in anomaly detection domain. Neural Comput. Applic. 33,13635–13646.

Elyan, E., Moreno-Garcia, C.F., Jayne, C., 2021. CDSMOTE: class decomposition and synthetic minority class oversampling technique for imbalanced-data classification. Neural Comput. Appl. 33 (7), 2839–2851.

Engelmann, J., Lessmann, S., 2021. Conditional wasserstein GAN-based oversampling of tabular data for imbalanced learning. Expert Syst. Appl. 174, 114582.

Farajzadeh-Zanjani, M., Hallaji, E., Razavi-Far, R., Saif, M., 2021. Generative adversarial dimensionality reduction for diagnosing faults and attacks in cyber-physical systems. Neurocomputing 440, 101–110.

Ferriyan, A., Thamrin, A.H., Takeda, K., Murai, J., 2021. Generating network intrusion detection dataset based on real and encrypted synthetic attack traffic. Appl. Sci. 11 (17), 7868.

Fox, J. (2021). Cybersecurity statistics for 2021. https://cobalt.io/blog/cybersecurity-statistics-2021 (Accessed on 16 August 2021).

Gabriel, A.J., Darwsih, A., Hassanien, A.E., 2021. Cyber security in the age of COVID-19. In: Digital Transformation and Emerging Technologies for Fighting COVID-19 Pandemic: Innovative Approaches. Springer, Cham, pp. 275–295.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A., 2017. Improved training of wasserstein gans. Adv. Neural. Inf. Process. Syst., *30*.

Gupta, N., Jindal, V., Bedi, P., 2022. CSE-IDS: using cost-sensitive deep learning and ensemble algorithms to handle class imbalance in network-based intrusion detection systems. Comput. Secur. 112, 102499.

Hasanin, T., Khoshgoftaar, T., 2018. The effects of random undersampling with simulated class imbalance for big data. In: *2018 IEEE International Conference on Information Reuse and Integrati*on (IRI). IEEE, pp. 70–79.

Huang, S., Lei, K., 2020. IGAN-IDS: an imbalanced generative adversarial network towards intrusion detection system in ad-hoc networks. Ad Hoc Netw. 105, 102177.

Khraisat, A., Gondal, I., Vamplew, P., Kamruzzaman, J., 2019. Survey of intrusion detection systems: techniques, datasets and challenges. Cybersecurity 2 (1), 1–22.

Kim, J.Y., Bu, S.J., Cho, S.B., 2018. Zero-day malware detection using transferred generative adversarial networks based on deep autoencoders. Inf. Sci. (Ny) 460, 83–102.

Kim, J., Jeong, K., Choi, H., Seo, K., 2020. Gan-based anomaly detection in imbalance problems. In: European Conference on Computer Vision, Cham. Springer, pp. 128–145.

Koroniotis, N., Moustafa, N., Sitnikova, E., Turnbull, B., 2019. Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: bot-iot dataset. Fut. Gen. Comput. Syst. 100, 779–796.

Lallie, H.S., Shepherd, L.A., Nurse, J.R., Erola, A., Epiphaniou, G., Maple, C., Bellekens, X., 2021. Cyber security in the age of covid-19: a timeline and analysis of cyber-crime and cyber-attacks during the pandemic. Comput. Secur. 105, 102248.

Lee, J., Park, K., 2021. GAN-based imbalanced data intrusion detection system. Pers. Ubiquitous Comput. 25 (1), 121–128.

Li, D., Li, Q., 2020. Adversarial deep ensemble: evasion attacks and defenses for malware detection. IEEE Trans. Inf. Forens. Secur. 15, 3886–3900.

Li, Y., Wang, Y., Hu, S., 2019. Online generative adversary network based measurement recovery in false data injection attacks: a cyber-physical approach. IEEE Trans. Ind. Inform. 16 (3), 2031–2043.

Li, X., Chen, W., Zhang, Q., Wu, L., 2020. Building auto-encoder intrusion detection system based on random forest feature selection. Comput. Secur. 95, 101851.

Liu, Z., Yin, X., 2021. LSTM-CGAN: towards generating low-rate DDoS adversarial samples for blockchain-based wireless network detection models. IEEE Access 9, 22616–22625.

Manship, R. The top 6 industries at risk for cyber attacks. RedTeam security threat prevention experts. https://www.redteamsecure.com/blog/the-top-6-industries-at-risk-for-cyber-attacks (Accessed on 16 August 2021).

Manzoor, I., Kumar, N., 2017. A feature reduced intrusion detection system using ANN classifier. Expert Syst. Appl. 88, 249–257.

McHugh, J., 2000. Testing intrusion detection systems: a critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln laboratory. ACM Trans. Inf. Syst. Secur. (TISSEC) 3 (4), 262–294.

Moti, Z., Hashemi, S., Karimipour, H., Dehghantanha, A., Jahromi, A.N., Abdi, L., Alavi, F., 2021. Generative adversarial network to detect unseen Internet of Things malware. Ad Hoc Netw. 122, 102591.

Moustafa, N., Slay, J., 2015. UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In: 2015 Military Communications and Information Systems Conference (MilCIS). IEEE, pp. 1–6.

Mozo, A., González-Prieto, Á., Pastor, A., Gómez-Canaval, S., Talavera, E., 2022. Synthetic flow-based cryptomining attack generation through generative adversarial networks. Sci. Rep. 12 (1), 1–27.

Nabe, C. (2021). Impact of COVID-19 on cybersecurity. https://www2.deloitte.com/ch/en/pages/risk/articles/impact-covid-cybersecurity.html (Accessed on 16 August 2021).

Nie, L., Wu, Y., Wang, X., Guo, L., Wang, G., Gao, X., Li, S., 2021. Intrusion detection for secure social internet of things based on collaborative edge computing: a generative adversarial network-based approach. IEEE Trans. Comput. Soc. Syst. 9 (1), 134–145.

Okereafor, K., 2021. Cybersecurity in the COVID-19 Pandemic. CRC Press.

Ring, M., Schlör, D., Landes, D., Hotho, A., 2019a. Flow-based network traffic generation using generative adversarial networks. Comput. Secur. 82, 156–172.

Ring, M., Wunderlich, S., Scheuring, D., Landes, D., Hotho, A., 2019b. A survey of network-based intrusion detection data sets. Comput. Secur. 86, 147–167.

Schwartz, H.A. Significant cyber incidents. Center for strategic & international studies. https://www.csis.org/programs/strategic-technologies-program/significant-cyber-incidents (Accessed on 16 August 2021).

Shawahna, A., Abu-Amara, M., Mahmoud, A.S., Osais, Y., 2018. EDoS-ADS: an enhanced mitigation technique against economic denial of sustainability (EDoS) attacks. IEEE Trans. Cloud Comput. 8 (3), 790–804.

Siniosoglou, I., Radoglou-Grammatikis, P., Efstathopoulos, G., Fouliras, P., Sarigiannidis, P., 2021. A unified deep learning anomaly detection and classification approach for smart grid environments. IEEE Trans. Netw. Serv. Manag. 18 (2), 1137–1151.

Sobers, R. (2022). Cybersecurity statistics and trends for 2022. https://www.varonis.com/blog/cybersecurity-statistics/ (Accessed on 16 August 2022).

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

Vinayakumar, R., Alazab, M., Soman, K.P., Poornachandran, P., Al-Nemrat, A., Venkatraman, S., 2019. Deep learning approach for intelligent intrusion detection system. IEEE Access 7, 41525–41550.

Vuttipittayamongkol, P., Elyan, E., 2020. Overlap-based undersampling method for classification of imbalanced medical datasets. In: IFIP International Conference on Artificial Intelligence Applications and Innovations, Cham. Springer, pp. 358–369.

Yan, Q., Wang, M., Huang, W., Luo, X., Yu, F.R., 2019. Automatically synthesizing DoS attack traces using generative adversarial networks. Int. J. Mach. Learn. Cybern. 10 (12), 3387–3396.

Yang, Y., Zheng, K., Wu, C., Yang, Y., 2019. Improving the classification effectiveness of intrusion detection by using improved conditional variational autoencoder and deep neural network. Sensors 19 (11), 2528.

Yu, Y., Tang, B., Lin, R., Han, S., Tang, T., Chen, M., 2019. CWGAN: conditional wasserstein generative adversarial nets for fault data generation. In: 2019 *IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, pp. 2713–2718.

Zhang, Y., Wang, J., Chen, B., 2020. Detecting false data injection attacks in smart grids: a semi-supervised deep learning approach. IEEE Trans. Smart Grid 12 (1), 623–634.

Zheng, M., Li, T., Zhu, R., Tang, Y., Tang, M., Lin, L., Ma, Z., 2020. Conditional Wasserstein generative adversarial network-gradient penalty-based approach to alleviating imbalanced data classification. Inf. Sci. (NY) 512, 1009–1023.

Zheng, M., Li, T., Zhu, R., Tang, Y., Tang, M., Lin, L., Ma, Z., 2020b. Conditional wasserstein generative adversarial network-gradient penalty-based approach to alleviating imbalanced data classification. Inf. Sci. (NY) 512, 1009–1023.

Zhou, Y., Cheng, G., Jiang, S., Dai, M., 2020. Building an efficient intrusion detection system based on feature selection and ensemble classifier. Comput. Netw. 174, 107247.

Zhu, T., Lin, Y., Liu, Y., 2017. Synthetic minority oversampling technique for multiclass imbalance problems. Pattern Recognit. 72, 327–340.

Vikash Kumar has received his M. Tech and Ph.D. degree from the Department of Computer Science and Engineering from National Institute of Technology Patna, Bihar (800005), India. He is currently working as an assistant professor in the department of computer science & Engineering, ITER, Sikasha O Anusandhan Deemed to be University, Bhubaneswar. His-research interest includes Intrusion Detection System, Network and Cyber security, Machine Learning and Deep Learning.

Ditipriya Sinha has received PhD degree in the Department of Computer Science and Technology, Indian Institute of Engineering Science and Technology (IIEST), Shibpur and Master of Technology from West Bengal University of Technology in the department of Software Engineering. She is the silver medalist during MTECH. She is presently serving as an assistant professor in the department of Computer Science and Engineering, National Institute of Technology Patna, Bihar (800005), India. She was an assistant professor in the department of Computer Science and Engineering, Birla Institute of Technology, Mesra. Her area of research is Mobile Ad-hoc Network, Wireless Sensor Network, Blockchain, Security and Scheduling algorithms.