Journal of Information Systems Engineering and Management
2025, 10(57s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Improving IDS Accuracy via XGBoost- Guided Tabular GANs: A Study on Synthetic Attack Data Generation

S. Ravisankar[1],Anjusree Krishnanunni[2] and Preethi Krishnan[3,*]
IT Faculty, Britts Imperial University College, Sharjah, UAE,
Email:s.sankarravi@gmail.com,Anjusree.k@nationalacademy.edu.in,preethikrish333@gmail.com

ARTICLE INFO	ABSTRACT
Received: 26 Dec 2024 Revised: 14 Feb 2025 Accepted: 22 Feb 2025	Intrusion detection systems (IDSs) are crucial components for safeguarding information systems against cyberattacks. Recently, machine learning (ML) has significantly enhanced the effectiveness of IDSs by analyzing patterns in network traffic from training data and applying this knowledge to predict new incoming data. However, training an effective ML model for IDSs requires collecting large datasets of normal and attacking samples. While normal samples can be gathered from the daily operation of network systems, attacking samples are much rarer and harder to collect. To mitigate this problem, numerous studies have been proposed to generate synthesized attacks. Previous studies are often based on training a generative adversarial network to generate synthesized attacks without considering the importance and relevance of the features in the dataset. In this paper, we propose a novel method for generating synthesized attacks based on training two advanced GAN models with weighting the importance of features in the dataset. The proposed methods are extensively evaluated on two IDS datasets. The experimental results demonstrate the effectiveness of the proposed methods in detecting both known and unknown attacks. **Keywords:** Generating Pseudo-attacks, Generative Mod- Adversarial Networks, Intrusion Detection

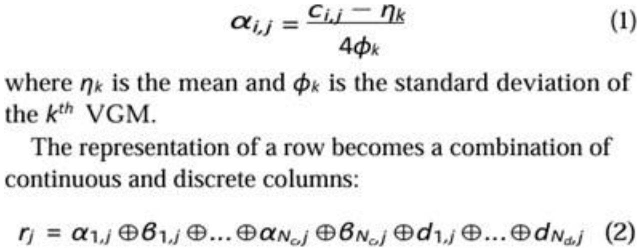
INTRODUCTION

Nowadays, Intrusion Detection Systems (IDSs) are widely used in many network systems to detect unexpected behaviors [1]. An IDS monitors computers and network systems to identify actions that threaten the systems or violate security policies and report them to system administrators. The main functions of an IDS are monitoring, alerting, and protecting against intrusions based on the network settings and configurations. IDSs mainly use two primary methods for intrusion detection: signature-based detection and anomaly-based detection [2]. Signature-based detection is designed to identify potential threats by comparing network traffic to known attack patterns. This method enables accurate detection and identification of known attacks. However, it is ineffective in detecting novel or unknown attacks. Conversely, anomaly-based intrusion detection is designed to identify both known and unknown attacks using machine learning techniques. Recently, machine learning methods have been widely applied to intrusion detection systems as a robust and accurate defense mechanism. To train an effective machine learning model for IDSs, it is important to collect a large dataset of both normal and abnormal/attacking samples. However, collecting high- quality datasets for IDSs is challenging due to privacy concerns. Moreover, while the normal samples can be easier to collect from the daily operation of systems, the abnormal or attacking samples are often much rarer. Thus, it is often more time-consuming and resource- consuming to gather abnormal samples. Several methods have been proposed to handle the problem of lacking abnormal samples by using Generative Adversarial Networks (GANs) [3] to generate synthesized data. For example, the studies in [4] [5] [6] [7] have attempted to augment the training dataset using GAN models for IDSs. These studies have shown that using synthetic data can mitigate the class imbalance problem in datasets and enhance the performance of IDSs. However, previous research did not pay attention to the role of different features of the datasets during generation. Thus, they are unable to generate data with specific characteristics. This paper aims to address the above issue by proposing a new method for generating synthesized attacks using two recently proposed models, i.e., CTGAN [8] and copula- GAN [9]. Our method is based on the important role of the features in the datasets. Specifically, we train a XG- Boost [10] model on each dataset and use the trained model to calculate the weight (or the important role) of each feature in the dataset. After that, the most important features are selected to encode into the CTGAN and Copula- GAN models. The resulting models are called xg-CTGAN and xg-Copula GAN, respectively. These models, i.e., xg- CTGAN and xg-Copula GAN, will specially focus on the important selected features during the training process. After training, they are used to generate synthesized attacks for a specific attack in each dataset. The synthesized attacks are then combined with the original attacks and the normal samples to form the augmented datasets. Three classifiers are trained on the augmented datasets. We evaluate the effectiveness of xg-CTGAN and xg-CopulaGAN on two IDSs bench-marking datasets: CICIDS2017 and UNSW- NB15. The experimental results show that xg-CTGAN and xg- CopulaGAN help classifiers achieve better results in detecting both known and unknown attacks compared to original datasets and some state-of-the-art generative models. The main contributions of our paper are as follows.

We propose a method to weigh the important role of features in datasets using XG Boost and select the most important features to generate synthesized attacks using CTGAN and Copula GAN. • We evaluate the effectiveness of the proposed methods on two IDSs bench-marking datasets and the experimental results show the superior performance of the proposed methods. The remainder of this paper is organized as follows. Section II presents two fundamental models of our paper: CTGAN and CopulaGAN Section III discusses related works, and Section IV presents a detailed description of our proposed method. Section V presents experimental settings. The experimental results and discussion are presented in Section VI. Finally, in Section VII, we conclude the paper and highlight future research directions.

BACKGROUNDS

This section briefly presents two models: CTGAN and CopulaGAN. They are two fundamental models of our proposed method. This model is specifically designed for generating synthetic tabular data. Specifically, the authors introduce the mode-specific normalization technique to overcome the non-Gaussian and multi- modal distribution issue in tabular data. Moreover, they also design a conditional generator to deal with the imbalanced discrete columns. Mode-specific normalization: Mode-specific normalization is a technique in CTGAN designed to address non-Gaussian and multi-modal distributions in tabular data. This technique ensures that features with complex distributions are appropriately normalized, allowing the model to learn better and generate higher-quality synthetic samples. In this approach, each column is processed independently. The value is represented by a one-hot encoded vector indicating the mode and a scalar value representing the value within that mode. For each continuous column, the Variational Gaussian Mixture (VGM) model is used for representation. The value Cij at row i and column j is represented as follows: vector, αi,j is a scalar value representing the actual value within the chosen mode



$$\alpha_{i,j} = \frac{c_{i,j} - \eta_k}{4\phi_k} \quad (1)$$

where η_k is the mean and ϕ_k is the standard deviation of the k^{th} VGM.

The representation of a row becomes a combination of continuous and discrete columns:

$$r_j = \alpha_{1,j} \oplus \beta_{1,j} \oplus \ldots \oplus \alpha_{N_c,j} \oplus \beta_{N_c,j} \oplus d_{1,j} \oplus \ldots \oplus d_{N_d,j} \quad (2)$$

where di,j representing the one-hot encoding of a discrete value.

Conditional Generator: The conditional generator is used to ensure that all values of a discrete feature are sampled evenly. Discrete columns often contain values with different frequencies. In other words, some values occur less frequently, while others have higher frequencies. This leads to the problem where GAN only learns the distribution of high-frequency values, ignoring the low-frequency values. To address this issue, the authors proposed a Conditional Generator to ensure that all values of the discrete columns are selected during the training process.

Assume that D1, D2, . . . , DN are the sets of discrete columns. Di = {di,j} is the set of possible values for the discrete column i (i ∈ [1, N]). Assume that k∗ is the selected value in the i∗-th discrete column (k∗ ∈ Di∗).

The Conditional Generator allows the creation of a conditional distribution of rows, specifically generating rows conditioned on the i∗-th column having the value k∗:

$$\hat{r} \sim P_G(\text{row} \mid D_{i^*} = k^*) \qquad (3)$$

During the training process, the Conditional Generator will learn the conditional distribution of real data, as expressed in equation 4:

$$P_G(\text{row} \mid D_{i^*} = k^*) = P(\text{row} \mid D_{i^*} = k^*) \qquad (4)$$

B. CopulaGAN: (4) CopulaGAN is a variant of CTGAN. Unlike CTGAN, which uses VGM to represent continuous columns, copula GAN uses Gaussian Copula [11] to transform continuous columns that do not follow a normal distribution into a normal distribution while preserving the relationships between columns. Assume Ci represents the i-th continuous column. The main idea of CopulaGAN includes two steps: First, convert each i-th continuous column from the input data into a normal distribution. Specifically, a Cumulative Distribution Function (CDF) corresponding to the distribution of each non-categorical variable is calculated. After that, the Inverse CDF is used to convert this value into a value that belongs to a normal distribution using the following equation:

$$Z_i = \Phi^{-1}(F_{C_i}(C_i)) \qquad (5)$$

Second, after obtaining the normal distributions Z for all continuous columns C, the transformed dataset is input to a CTGAN model for training. In summary, CopulaGAN is extended from CTGAN by replacing the mode-specific normalization in CTGAN with the Copula transformation.

RELATED WORKS

This section briefly reviews the previous studies on improving GAN and their application to generating data in IDSs. GAN, proposed by Goodfellow et al. [3], is marked as one of the most significant breakthroughs in this field of generative models. However, GAN has some limitations, such as instability, mode collapse, and gradient vanishing during training. Subsequently, various improvements of GANs have been proposed.

WGAN (Wasserstein Generative Adversarial Net- work) [12] is an adaptation of the original GAN model designed to overcome problems like instability and mode collapse that are often seen in traditional GANs. WGANs use the Wasserstein distance in its loss function results in more stable training. Moreover, WGAN also introduces weight clipping to ensure Lipschitz continuity, a requirement for the Wasserstein distance. WGAN-GP (Wasserstein Generative Adversarial Network with Gradient Penalty) [13] is an improved variant of WGAN to address the shortcomings of weight clipping in the WGAN. WGAN-GP modifies the Lipschitz continuity, leading to more stable and efficient training. WGAN-GP replaces weight clipping with a gradient penalty, which penalizes the discriminator for deviating from a gradient norm, maintaining Lipschitz continuity more flexibly and effectively.

Recently, CTGAN [8] was specially designed to address the challenges in synthesizing tabular data. CTGAN applies different normalization methods for each column in the data. This helps overcome the limitations of traditional normalization methods, which often assume that the data follows a Gaussian distribution. Additionally, CTGAN uses a conditional generator to address the issue of data imbalance in categorical columns. The conditional generator enables the creation of conditional vectors, allowing CTGAN to explore all possible discrete values evenly. CopulaGAN [9] is a variant of CTGAN focused on modeling the dependency between variables. This model combines CTGAN with copula methods, specifically Gaussian Copula, to handle non-standard distributions and complex relationships between variables.

There are also a large number of research focusing on applying GAN variants to intrusion detection. Shahriar et al. [14] proposed a generative adversarial network (GAN) based intrusion detection system (G- IDS), where a GAN is used to generate synthetic samples. G-IDS also fixes the difficulties of imbalanced or missing data problems. The experiments on the NSLKDD dataset show that the proposed model performs much better in attack detection than a standalone IDS. Bourou et al. [15] evaluated the effectiveness of tabular data synthesis using GANs on the IDS dataset. They applied the CTGAN, CopulaGAN, and table GAN models to the NSL-KDD dataset and indicated that synthetic data can be used to train network attack detection models. Dina et al [16] addressed the data imbalanced problem in IDSs using synthetic data (CTGAN). Halvorsen et al. [17] reviewed the application of generative machine learning in intrusion detection. The authors analyzed the application of various generative models including GANs, VAEs, etc. to different problems in IDSs like false data attacks, evasion attacks, exploratory attacks, and dealing with unbalanced datasets. Overall, the previous research provides evidence for the usefulness of GANs in generating synthetic data for IDSs. However, these methods do not consider the important role of features in the datasets during the training process. In this paper, we propose a novel method to generate synthesized attacks by emphasizing the role of features related to the attacks. A detailed description of our method will be presented in Section IV.

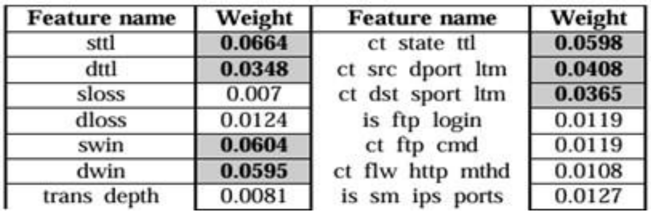
PROPOSED APPROACH

This section presents our proposed method in detail. We first present a novel method for evaluating the important role of features in datasets. After that, we highlight the overall architecture of our system for intrusion detection.

A.Selecting Important Features

In IDSs, the features of datasets often have different roles. Usually, each feature presents the characteristic of one or some specific attacks. Thus, when generating synthesized attacks, it is important to focus more on some features than others. To achieve this goal, we propose a method to select important features based on the XGBoost model. XGBoost (Extreme Gradient Boosting) [10] is a machine learning algorithm based on the gradient boosting framework. It is designed for efficiency, scalability, and performance, making it suitable for a wide range of machine learning tasks, particularly in structured/tabular data. Particularly, XGBoost can be used to rank and select features based on their importance to the model.

TABLE I
WEIGHTS AND SELECTED CATEGORICAL FEATURES IN UNSW-NB15



Feature name	Weight	Feature name	Weight
sttl	**0.0664**	ct state ttl	**0.0598**
dttl	**0.0348**	ct src dport ltm	**0.0408**
sloss	0.007	ct dst sport ltm	**0.0365**
dloss	0.0124	is ftp login	0.0119
swin	**0.0604**	ct ftp cmd	0.0119
dwin	**0.0595**	ct flw http mthd	0.0108
trans depth	0.0081	is sm ips ports	0.0127

First, we train the XGBoost model on all the features in the training set. After training, we extract the weight of each feature. We focus on the weights of the categorical features since the CTGAN and CopulaGAN models only support inputting one-hot vectors of categorical features. We select the set of categorical features D∗ with the weight greater than the average weight of all features. Table I presents the features selected in the UNSW-NB15 dataset 1. After that, the selected categorical features will be used to initialize the input condition vector for the CTGAN and CopulaGAN algorithms. The detailed steps of our method are described in Figure 1.

B. System Architecture

The architecture of the IDS that uses our proposed method for generating synthesized attacks is presented in Figure 2. This model consists of three phases: synthesis phase, training phase, and inference phase. In the synthesis phase, we first train the CTGAN and CopulaGAN models with selected important features. We name these models as xg-CTGAN and xg-CopulaGAN. After training, we use these models to synthesize attacking samples. In the training phase, the synthesized samples are combined with the original samples to form a new augmented dataset. Detective models, i.e., classifiers, are trained on the augmented dataset. In the inference phase, test data is input into the trained classifiers to detect intrusions.

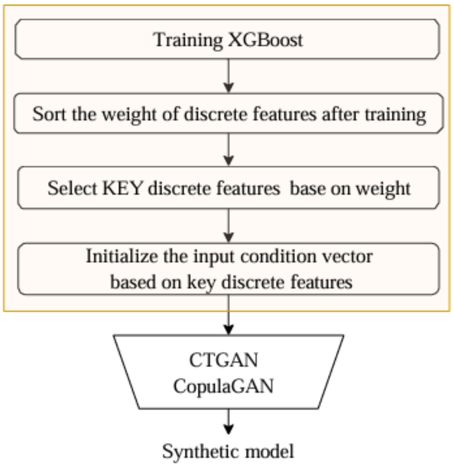

Training XGBoost
Sort the weight of discrete features after training
Select KEY discrete features base on weight
Initialize the input condition vector based on key discrete features
CTGAN
CopulaGAN
Synthetic model

Fig. 1. Training procedure with key discrete feature selection

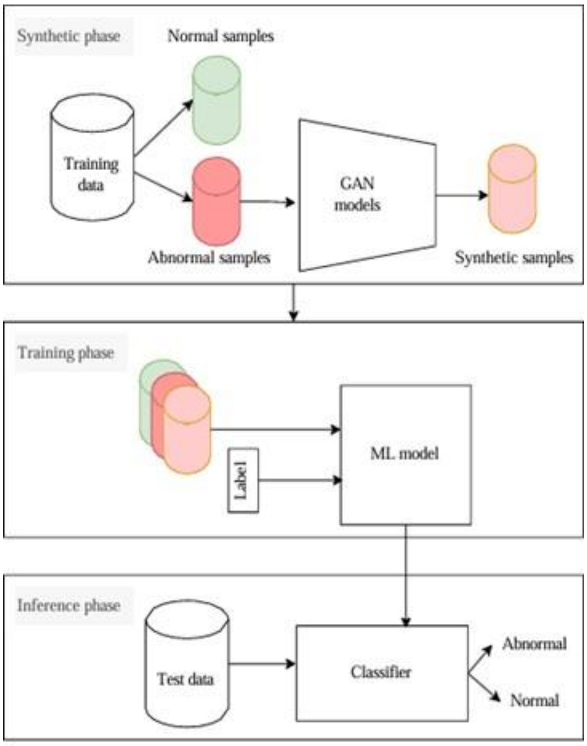


Fig. 2. Anomaly Detection Process

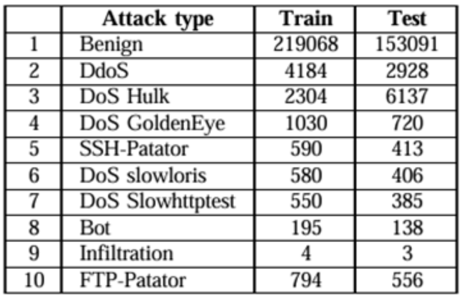
EXPERIMENTS

This section presents the data sets used in the experiments, the performance metrics, and the experimental settings.

A . Datasets

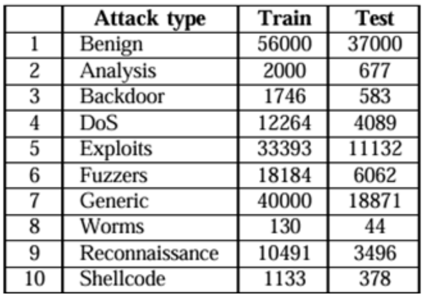
We tested the proposed methods on two datasets including CICIDS-2017, UNSW-NB15.

TABLE II
TYPES OF ATTACKS IN THE CIC-IDS2017 DATASET



	Attack type	Train	Test
1	Benign	219068	153091
2	DdoS	4184	2928
3	DoS Hulk	2304	6137
4	DoS GoldenEye	1030	720
5	SSH-Patator	590	413
6	DoS slowloris	580	406
7	DoS Slowhttptest	550	385
8	Bot	195	138
9	Infiltration	4	3
10	FTP-Patator	794	556

TABLE III
TYPES OF ATTACKS IN THE UNSW DATASET



	Attack type	Train	Test
1	Benign	56000	37000
2	Analysis	2000	677
3	Backdoor	1746	583
4	DoS	12264	4089
5	Exploits	33393	11132
6	Fuzzers	18184	6062
7	Generic	40000	18871
8	Worms	130	44
9	Reconnaissance	10491	3496
10	Shellcode	1133	378

The CICIDS-2017 dataset [18] includes both benign traffic and traffic from common attacks, like real- world data (PCAPs). The dataset is the result of network traffic analysis using CICFlowMeter, with flows labeled based on timestamp, source and destination IP, source and destination ports, protocol, and attack. It also includes definitions for the extracted features. The number of attack types in the dataset is described in Table II.

UNSW-NB15 [19] is a dataset collected by the Cyber Range Lab of the Australian Centre for Cyber Security (ACCS). The authors created a combination of realistic modern normal activities and synthetic attack behaviors using the IXIA PerfectStorm tool. This dataset includes 42 features and comprises nine attack classes and one normal class. The number of attack types in the dataset is described in Table III.

B. Performance Metrics

We use three popular metrics including Precision, Recall, and F1-score to evaluate the effectiveness of xg- CTGAN and xg-CopulaGAN. Precision: Precision calculates the ratio of correctly identified anomalies, known as True Positives (TP), to the total number of anomalies detected. A high Precision score signifies that the system is highly accurate in identifying abnormal cases. The formula for calculating Precision is presented in Eq. 6.

$$Precision = \frac{TP}{TP + FP}. \qquad (6)$$

Recall: Recall measures the ratio of the number of correctly detected anomalies, i.e., TP, to the total number of true anomalies. A high Recall value indicates that the system has good coverage of anomalous samples. The formula to calculate Recall is presented in Eq. 7 following:

$$Recall = \frac{TP}{TP + FN}. \qquad (7)$$

F1-score: The F1-score is a metric that combines Precision and Recall providing a single measure of a model's performance. This metric is especially useful when the class distribution is imbalanced. It is defined as the harmonic mean of Precision and Recall, and it is given by the formula in Eq. 8:

$$F1 - score = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \qquad (8)$$

C. Experimental Settings

We focus our research on investigating the effectiveness of the synthesized data in detecting both known and unknown attacks. For this, we only generate synthesized attacks for one type of attack in each dataset. Specifically, for CIDIDS2017, we generate 50000 synthesized samples for the DDoS attack and for UNSW- NB15, 50000 samples of the backdoor attack are generated.

We divided our experiments into two sets. The first set is to evaluate the effectiveness of the proposed method in detecting the attacks of the same class with the synthesized data. In this setting, one type of attack is synthesized and then they are combined with the original attack of the same type and the normal data to form the augmented dataset. The augmented datasets are then used to train the classifiers. The testing set contains only the normal and the attacks of the same type as the attacks in the training set. The second set is to evaluate their ability to detect new/unknown attacks. In the second setting, the training set is formed similarly to the first setting, however, the testing set contains all types of attacks in the dataset. We evaluate the effectiveness of xg-CTGAN and CopulaGAN using three classifiers including Random Forest RF, Support Vector Machines – SVM, and Multilayer Perceptron Classifier – MLP. The default settings of these classifiers in Scikit-Learn are selected. We also compare the data generated by xg-CTGAN and xg- CopulaGAN with the original dataset and the data generated by WGAN, WGAN-GP, CTGAN, and CopulaGAN.

RESULTS AND DISCUSSION

This section discusses the results of the experiments in two settings presented in Section V. We will use the F1- score as the main indicator for the performance of the tested methods. The results of Precision and Recall are used for reference only since they only present a partial picture of the effectiveness of each method.

A . Detecting the Known Attacks

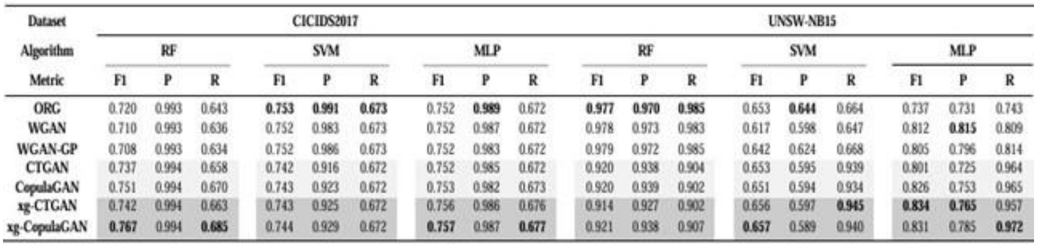
The results of three classifiers, RF, SVM, and MLP, on two tested datasets corresponding to different data synthesized models, are presented in Table IV.First, this table shows that data synthesized models often help the classifiers achieve better results compared to the original data. This is even more impressive when using CTGAN and CopulaGAN models. For example, the F1- score of RF on CICIDS2027 when trained on the original data is only 0.720 while these values of CTGAN and CopulaGAN are 0.738 and 0.751, respectively. Second, the proposed methods for generating data using important features, i.e., xg-CTGAN and xg-CopulaGAN, yield higher results compared to CTGAN and CopulaGAN. In particular, the xg-CopulaGAN model is better than CopulaGAN on both datasets. This demonstrates that the proposed methods for generating synthesized attacks that focus on important features significantly improve the performance of classification algorithms.

B. Ability to Detect New

Attacks Table V presents the results when the classifiers are used to detect new/unknown attacks. There are three interesting results observed from this table. First, detecting new attacks is much harder than detecting known attacks. Thus, the accuracy of all classifiers on all tested datasets is much lower than the corresponding values in Table IV. For instance, the F1-score of RF on CICIDS2017 when trained on original data (ORG) is only 0.479 while this value in Table IV is 0.720. Second, generating synthesized data supports classifiers that leverage their ability in detecting unknown attacks on most configurations. Specifically, the F1-score of SVM on CICIDS2017 when trained on the augmented data of CTGAN is 0.720 and this value is significantly higher than the value on the original data at 0.632. Last, our proposed methods for generating data usually help classifiers achieve the best results. For instance, the F1-score of SVM on UNSW-NB15 trained on the dataset of xg-CopulaGAN is 0.741 and this value is the highest one among all tested configurations. Overall, Table V shows that the data generated by xg-CTGAN and xg-CopulaGAN not only improves the performance of classifiers in general, but it also helps to improve the accuracy of classifiers in detecting unknown attacks.

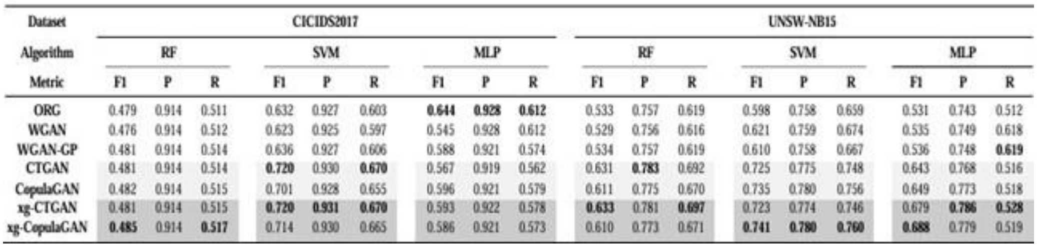
CONCLUSIONS

TABLE IV
F1-SCORE, PRECISION AND RECALL OF THREE CLASSIFIERS IN DETECTING KNOWN ATTACKS



Dataset	CICIDS2017									UNSW-NB15								
Algorithm	RF			SVM			MLP			RF			SVM			MLP		
Metric	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R
ORG	0.720	0.993	0.643	**0.753**	**0.991**	**0.673**	0.752	**0.989**	0.672	**0.977**	**0.970**	**0.985**	0.653	**0.644**	0.664	0.737	0.731	0.743
WGAN	0.710	0.993	0.636	0.752	0.983	0.673	0.752	0.987	0.672	0.978	0.973	0.983	0.617	0.598	0.647	0.812	**0.815**	0.809
WGAN-GP	0.708	0.993	0.634	0.752	0.986	0.673	0.752	0.983	0.672	0.979	0.972	0.985	0.642	0.624	0.668	0.805	0.796	0.814
CTGAN	0.737	0.994	0.658	0.742	0.916	0.672	0.752	0.985	0.672	0.920	0.938	0.904	0.653	0.595	0.939	0.801	0.725	0.964
CopulaGAN	0.751	0.994	0.670	0.743	0.923	0.672	0.753	0.982	0.673	0.920	0.939	0.902	0.651	0.594	0.934	0.826	0.753	0.965
xg-CTGAN	0.742	0.994	0.663	0.743	0.925	0.672	0.756	0.986	0.676	0.914	0.927	0.902	0.656	0.597	**0.945**	**0.834**	**0.765**	0.957
xg-CopulaGAN	**0.767**	0.994	**0.685**	0.744	0.929	0.672	**0.757**	0.987	**0.677**	0.921	0.938	0.907	**0.657**	0.589	0.940	0.831	0.785	**0.972**

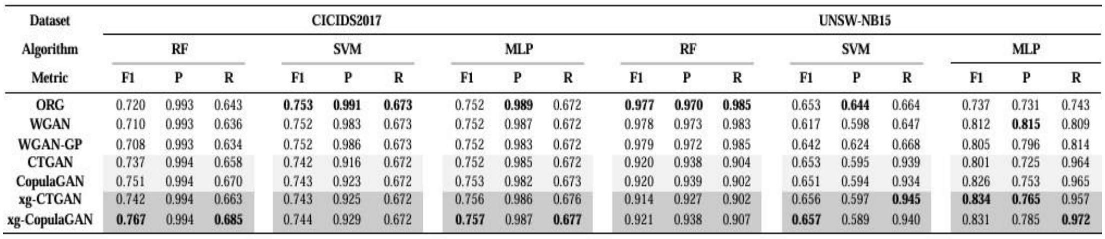
TABLE V
F1-SCORE, PRECISION AND RECALL OF THREE CLASSIFIERS IN DETECTING UNKNOWN ATTACKS



Dataset	CICIDS2017									UNSW-NB15								
Algorithm	RF			SVM			MLP			RF			SVM			MLP		
Metric	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R
ORG	0.479	0.914	0.511	0.632	0.927	0.603	**0.644**	**0.928**	**0.612**	0.533	0.757	0.619	0.598	0.758	0.659	0.531	0.743	0.512
WGAN	0.476	0.914	0.512	0.623	0.925	0.597	0.545	0.928	0.612	0.529	0.756	0.616	0.621	0.759	0.674	0.535	0.749	0.618
WGAN-GP	0.481	0.914	0.514	0.636	0.927	0.606	0.588	0.921	0.574	0.534	0.757	0.619	0.610	0.758	0.667	0.536	0.748	**0.619**
CTGAN	0.481	0.914	0.514	**0.720**	0.930	**0.670**	0.567	0.919	0.562	0.631	**0.783**	0.692	0.725	0.775	0.748	0.643	0.768	0.516
CopulaGAN	0.482	0.914	0.515	0.701	0.928	0.655	0.596	0.921	0.579	0.611	0.775	0.670	0.735	0.780	0.756	0.649	0.773	0.518
xg-CTGAN	0.481	0.914	0.515	**0.720**	**0.931**	**0.670**	0.593	0.922	0.578	**0.633**	0.781	**0.697**	0.723	0.774	0.746	0.679	**0.786**	**0.528**
xg-CopulaGAN	**0.485**	0.914	**0.517**	0.714	0.930	0.665	0.586	0.921	0.573	0.610	0.773	0.671	**0.741**	**0.780**	**0.760**	**0.688**	0.779	0.519

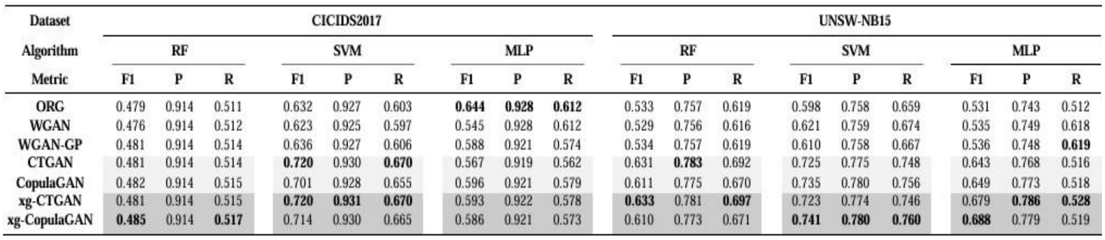
In this paper, we proposed two novel methods for generating synthesized attacks to augment training datasets in IDSs. Our proposed methods are based on weighing the role of features in the datasets and selecting the most important features to be input to two well-known models, CTGAN and CopulaGAN. For each dataset, we generate synthesized data for one specific attack and the generated data is then combined with the original dataset to form the augmented datasets. The experiments are conducted on two popular IDS benchmarking datasets, including CICIDS2017 and UNSW-NB15. The results show that the classifiers trained with synthesized data from our proposed methods enhance accuracy in detecting both known and unknown attacks. There are several future research directions raised from our paper. First, we want to focus on generating attacking datasets that are based on continuous features. Second, we would like to conduct experiments on a wider range of problems.

TABLE IV
F1-SCORE, PRECISION AND RECALL OF THREE CLASSIFIERS IN DETECTING KNOWN ATTACKS



Dataset	CICIDS2017									UNSW-NB15								
Algorithm	RF			SVM			MLP			RF			SVM			MLP		
Metric	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R
ORG	0.720	0.993	0.643	**0.753**	**0.991**	**0.673**	0.752	**0.989**	0.672	**0.977**	**0.970**	**0.985**	0.653	**0.644**	0.664	0.737	0.731	0.743
WGAN	0.710	0.993	0.636	0.752	0.983	0.673	0.752	0.987	0.672	0.978	0.973	0.983	0.617	0.598	0.647	0.812	**0.815**	0.809
WGAN-GP	0.708	0.993	0.634	0.752	0.986	0.673	0.752	0.983	0.672	0.979	0.972	0.985	0.642	0.624	0.668	0.805	0.796	0.814
CTGAN	0.737	0.994	0.658	0.742	0.916	0.672	0.752	0.985	0.672	0.920	0.938	0.904	0.653	0.595	0.939	0.801	0.725	0.964
CopulaGAN	0.751	0.994	0.670	0.743	0.923	0.672	0.753	0.982	0.673	0.920	0.939	0.902	0.651	0.594	0.934	0.826	0.753	0.965
xg-CTGAN	0.742	0.994	0.663	0.743	0.925	0.672	0.756	0.986	0.676	0.914	0.927	0.902	0.656	0.597	**0.945**	**0.834**	**0.765**	0.957
xg-CopulaGAN	**0.767**	0.994	**0.685**	0.744	0.929	0.672	**0.757**	0.987	**0.677**	0.921	0.938	0.907	**0.657**	0.589	0.940	0.831	0.785	**0.972**

TABLE V
F1-SCORE, PRECISION AND RECALL OF THREE CLASSIFIERS IN DETECTING UNKNOWN ATTACKS



Dataset	CICIDS2017									UNSW-NB15								
Algorithm	RF			SVM			MLP			RF			SVM			MLP		
Metric	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R
ORG	0.479	0.914	0.511	0.632	0.927	0.603	**0.644**	**0.928**	**0.612**	0.533	0.757	0.619	0.598	0.758	0.659	0.531	0.743	0.512
WGAN	0.476	0.914	0.512	0.623	0.925	0.597	0.545	0.928	0.612	0.529	0.756	0.616	0.621	0.759	0.674	0.535	0.749	0.618
WGAN-GP	0.481	0.914	0.514	0.636	0.927	0.606	0.588	0.921	0.574	0.534	0.757	0.619	0.610	0.758	0.667	0.536	0.748	**0.619**
CTGAN	0.481	0.914	0.514	**0.720**	0.930	**0.670**	0.567	0.919	0.562	0.631	**0.783**	0.692	0.725	0.775	0.748	0.643	0.768	0.516
CopulaGAN	0.482	0.914	0.515	0.701	0.928	0.655	0.596	0.921	0.579	0.611	0.775	0.670	0.735	0.780	0.756	0.649	0.773	0.518
xg-CTGAN	0.481	0.914	0.515	**0.720**	**0.931**	**0.670**	0.593	0.922	0.578	**0.633**	0.781	**0.697**	0.723	0.774	0.746	0.679	**0.786**	**0.528**
xg-CopulaGAN	**0.485**	0.914	**0.517**	0.714	0.930	0.665	0.586	0.921	0.573	0.610	0.773	0.671	**0.741**	**0.780**	**0.760**	**0.688**	0.779	0.519

Finally, we plan to carry out a deeper analysis of the synthesized data of xg-CTGAN and xg CopulaGAN to better understand its characteristics.

ACKNOWLEDGMENT

This research was funded by the Military Technical Academy through the Academy-level scientific and techno- logical research mission titled ”Research and Application of Techniques for Generating Synthetic Anomalous Data”.

REFERENCES

[1] Arash Heidari and Mohammad Ali Jabraeil Jamali. Internet of things intrusion detection systems: a comprehensive review and future directions. Cluster Computing, 26(6):3753–3780, 2023.

[2] Oluwadamilare Harazeem Abdulganiyu, Taha Ait Tchakoucht, and Yakub Kayode Saheed. A systematic literature review for network intrusion detection system (ids). International journal of information security, 22(5):1125–1162, 2023.

[3] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. Advances in neural information pro- cessing systems, 27, 2014.

[4] Milad Salem, Shayan Taheri, and Jiann Shiun Yuan. Anomaly gen- eration using generative adversarial networks in host-based intrusion detection. In 2018 9th IEEE Annual Ubiquitous Computing, Elec- tronics & Mobile Communication Conference (UEMCON), pages 683–687. IEEE, 2018.

[5] Tim Merino, Matt Stillwell, Mark Steele, Max Coplan, Jon Patton, Alexander Stoyanov, and Lin Deng. Expansion of cyber-attack data from unbalanced datasets using generative adversarial networks. Software Engineering Research, Management and Applications, pages 131–145, 2020.

[6] Md Hasan Shahriar, Nur Imtiazul Haque, Mohammad Ashiqur Rah- man, and Miguel Alonso. G-ids: Generative adversarial networks assisted intrusion detection system. In 2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC), pages 376–385. IEEE, 2020.

[7] Ibrahim Yilmaz, Rahat Masum, and Ambareen Siraj. Addressing imbalanced data problem with generative adversarial network for intrusion detection. In 2020 IEEE 21st international conference on information reuse and integration for data science (IRI), pages 25–30. IEEE, 2020.

[8] Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Modeling tabular data using conditional gan. Advances in neural information processing systems, 32, 2019.

[9] Synthetic data vault. Retrieved July 14, 2024, from https://sdv.dev/SDV/.

[10] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, pages 785– 794, 2016.

[11] Sanket Kamthe, Samuel Assefa, and Marc Deisenroth. Copula flows for synthetic data generation. arXiv preprint arXiv:2101.00598, 2021.

[12] Martin Arjovsky, Soumith Chintala, and Le´on Bottou. Wasserstein generative adversarial networks. International conference on machine learning, pages 214–223. PMLR, 2017.

[13] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Du- moulin, and Aaron C Courville. Improved training of wasserstein gans. Advances in neural information processing systems, 30, 2017.

[14] Md Hasan Shahriar, Nur Imtiazul Haque, Mohammad Ashiqur Rah- man, and Miguel Alonso. G-ids: Generative adversarial networks assisted intrusion detection system. In 2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC), pages 376–385, 2020.

[15] Stavroula Bourou, Andreas El Saer, Terpsichori-Helen Velivassaki, Artemis Voulkidis, and Theodore Zahariadis. A review of tabular data synthesis using gans on an ids dataset. Information, 12(09):375, 2021.

[16] Ayesha Siddiqua Dina, AB Siddique, and D Manivannan. Effect of balancing data using synthetic data on the performance of machine learning classifiers for intrusion detection in computer networks. IEEE Access, 10:96731–96747, 2022.

[17] James Halvorsen, Clemente Izurieta, Haipeng Cai, and Assefaw Gebremedhin. Applying generative machine learning to intrusion detection: A systematic mapping study and review. ACM Computing Surveys, 56(10):1–33, 2024.

[18] Iman Sharafaldin, Arash Habibi Lashkari, and Ali A Ghorbani. A detailed analysis of the cicids2017 data set. In Information Systems Security and Privacy: 4th International Conference, ICISSP 2018, Funchal-Madeira, Portugal, January 22-24, 2018, Revised Selected Papers 4, pages 172–188. Springer, 2019.

[19] Nour Moustafa and Jill Slay. Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set). In 2015 military communications and information systems conference (MilCIS), pages 1–6. IEEE, 2015.