Assessing the Quality of Differentially Private Synthetic Data for Intrusion Detection

Md Ali Reza Al Amin[1(✉)], Sachin Shetty[1], Valerio Formicola[2], and Martin Otto[3]

[1] Old Dominion University, Norfolk, VA 23508, USA
{malam002,sshetty}@odu.edu
[2] California Polytechnic State University, Pomona, California, USA
vformicola@cpp.edu
[3] Siemens Technology US, Princeton, NJ 08540, USA
m.otto@siemens.com

Abstract. Supervised learning is effectively adopted in Network Intrusion Detection Systems (IDS) to detect malicious activities in Information Technology (IT) and Operation Technology (OT). Sharing high-quality network data among cyber-security practitioners increases the chance of detecting new threat campaigns by analyzing updated traffic features. As data sharing brings privacy concerns, Differential-Privacy (DP) has emerged as a promising approach to performing privacy-preserving analytics. This paper presents an approach to generating high-quality synthetic network features using a differentially private Generative Adversarial Network (DP-GAN) based on the DoppleGANger https://github.com/fjxmlzn/DoppelGANger toolset. We assess the classification performance of several machine learning (ML) models on a privacy-preserved synthetic dataset derived from the NSL-KDD intrusion dataset. Experiments show ML algorithms achieve high classification accuracy on the synthetic data (95.95%) with a low privacy budget ($\varepsilon = 6.73$), i.e., low success rates for membership inference attacks. Hence, DP-GAN models offer a promising tool for sharing traffic features with bounded loss of privacy.

Keywords: Intrusion detection system · Differential privacy · Generative adversarial networks · Data sharing

1 Motivation

With the increasing adoption of Information Technology (IT) and Operation Technology (OT), Intrusion Detection System (IDS) is one of the most critical defensive mechanisms against cyber-attacks with potential impact on the cyber-physical system. Network IDS detects malicious or anomalous activities

V. Formicola—This work has been performed while at Siemens Technology US.

F. Li et al. (Eds.): SecureComm 2022, LNICST 462, pp. 473–490, 2023.
https://doi.org/10.1007/978-3-031-25538-0_25

within a network domain by analyzing network traffic characteristics (features). Detecting if network traffic is malicious or benign, or determining the attack category, is a classification problem. Several supervised machine learning models have been widely used to train IDS and improve detection accuracy with good success on known attack methods. However, as attackers develop new techniques, for example, developing and using variants of known attack tools, models have to be re-trained against new features. Updating a data-driven network IDS is more effective if information and data are shared among cyber-security practitioners in a timely manner. As a matter of fact, data sharing is very controlled and limited, despite the benefit, due to privacy concerns in the exposure of sensitive information.

Most anonymization techniques require a subset of real data to be shared among security practitioners. However, for some domains sharing real data with remote entities imposes a security threat. For example, intrusion detection datasets contain real attack signatures and sensitive information (i.e., source and destination IP addresses, port numbers, etc.). During data sharing, attackers can use the attack signature information to learn how to bypass the detection if the real data gets compromised. At the same time, large network-level intrusion data under attack scenarios can help to build good machine learning-based intrusion detection systems. However, the lack of such attack datasets has significantly hampered data-driven research. Sometimes, it is not feasible to share data among different teams, even within the organization. To guarantee no part of the attack datasets is being shared outside the organization's private network, we need an alternative method to sharing real data that can be used to build a robust machine learning model.

In this work, we leverage a tool, DoppleGANger (DG) [13] to build the synthetic data generation framework and investigate the synthetic data utility -i.e., classification of malicious traffic - while protecting the privacy of information contained in a dataset. DG is developed to generate synthetic networked time-series data using generative adversarial networks (GANs) while improving the data fidelity, i.e., long-term dependencies, complex multidimensional relationships, mode collapse. DG claims to achieve 43% better fidelity than other baseline models. DG also tackles the mode collapse issue of GANs by developing a custom auto-normalization heuristic method. Previous studies [5] have proposed synthetic data generation using Decision Trees, Random Forest, etc., by keeping the statistical properties like mean or median close to the original dataset. However, it is also important to test the machine learning model's performance (i.e., classification accuracy) on the original dataset when the model is trained on the synthetic dataset. One promising way to generate synthetic data that resembles original data for analytical tasks is to use GANs. GANs are neural networks that use random noise as input and generate realistic data samples. GANs have been widely used to generate synthetic data and translation in image and text data [2,9]. However, GAN-based models are prone to Membership Inference Attacks (MIA). In MIA, the attacker aims to identify whether a specific record was used to train the model. Moreover, GAN-based generation does not allow the user to quantify and assess the level of privacy achieved. To alleviate that issue, we

apply the differential privacy on GAN, DP-GAN, generated synthetic data to make the privacy guarantee and make the machine learning model robust against MIA. The DP-GAN model consists of two networks, a generator and a discriminator, which can be modeled based on the application domain. Long Short Term Memories (LSTM) is used inside the generator to model continuous data, and Multilayer Perceptron (MLP) is used to model discrete data. To achieve the differential privacy guarantee, we used the Differentially Private Stochastic Gradient Descent (DP-SGD), proposed by [1], to train the discriminator and the Adam optimizer to train the generator.

We use a well-known Intrusion Detection Dataset, NSL-KDD, to conduct the experiments. To date, NSL-KDD has still been considered an intrusion detection benchmark because of its diverse attacks groups [18]. In summary, we provide the following contributions:

- Generate differentially private synthetic intrusion detection dataset from NSL-KDD while maintaining high accuracy for classification. Further, we assess the differential privacy achieved in the synthetic intrusion detection dataset. Even if GAN-based approaches are well studied, no quantification has been done so far regarding differential privacy and against membership inference attacks. Therefore, our investigation on differentially private intrusion detection, where we retain 95% classification accuracy, supports the research on synthetic data in the cyber-security community.
- We perform an assessment of detection accuracy by following a use case scenario where data is trained on a differentially private synthetic dataset and tested on the original dataset for validation. As a result, we achieve 90% accuracy in detecting malicious traffic from benign traffic.
- We find the parameters of DP-GAN to generate synthetic data that achieve a trade-off between privacy and accuracy for intrusion detection.

The rest of the paper is as follows: in Sect. 2, we describe the related work, Sect. 3 describes the Privacy-Preserving framework, Sect. 4 explains how we prepare the dataset and privacy we are protecting, and lastly, in Sect. 5, experimental results are presented.

2 Related Work

Privacy-preserving data sharing has been widely discussed in the past. However, organizations are still skeptical about sharing their own data for use in research. To improve the intrusion detection accuracy, a large volume of network data is needed. Previous technique for IDS like Snort [3] works with detection rule for known attack. The main drawback for the rule based method is that it does not perform well for novel attacks. This is why ML based methods are currently being used for the automated rule generation by the machine. Previous efforts [14,21] in intrusion detection heavily rely on large volumes of network data. Sharing these data with the research community is not practical as the dataset contains sensitive information that impose privacy guarantee.

There has been some work done on improving the detection accuracy using GAN using the dataset we have used in this paper. Authors in [12] propose a framework based on GANs to generate adversarial samples to improve intrusion detection. NSL-KDD dataset is used to test the feasibility of the model. The authors in [17] proposed a generative adversarial network (GAN) based intrusion detection system (G-IDS), where GAN generates synthetic samples, and IDS gets trained on them along with the original ones. GAN based IDS work focused on generating adversarial samples but does not guarantee the privacy. Several other approaches as in [8,16,22] use GANs to generate synthetic datasets for ad-hoc and Industrial Internet of Things (IIoT) networks. These models are focused on solving the imbalance problems in the intrusion detection dataset. Compared to these works, our work focuses on generating the synthetic intrusion detection dataset and assessing privacy while maintaining high-level classification accuracy.

The authors in [11], propose a framework to generate privacy-preserving synthetic data suitable for release for analytical purposes. To ensure the privacy, the principal of t-closeness is applied GAN model. PATE-GAN [10] is the Private Aggregation of Teacher Ensembles (PATE) framework which applies to the GANs. Another differentially private generative adversarial network approach described in [19] achieved differential privacy in GANs by adding carefully designed noise to gradients during learning procedure. The authors used MNIST and MIMIC-III dataset to do evaluation. However, to the best of our knowledge so far DP-GAN is not directly applied to the Network Intrusion Detection System dataset NSL-KDD. We also show that, DP-GAN is robust than the GAN to defend against membership inference attacks.

3 Privacy Preserving Framework

In this section, we elaborate on the design of privacy-preserving framework, a differentially private generative adversarial network, to mitigate privacy exposure and maintain desirable utility in the generated intrusion detection data. In our proposed approach, we first train a DG model using real data. After the training, the generator in the DG model can generate a sample dataset which can be used instead of real dataset. Then, differential privacy is applied on the GAN generated synthetic dataset to meet the privacy guarantee and defend against MIA. We elaborate the detail of data generation and preserving privacy in our approach further in this section.

3.1 Generation Using DG

GANs are a data-driven generative modeling technique that takes training data samples as an input (original data) and outputs a model that generates new samples from the same distribution of the input. A GAN consists of a generator G and a discriminator D. First, the generator maps the noise vector to samples and generates plausible data. Then, the discriminator is trained by taking samples as input and classifying those samples as fake or real. More precisely, GANs

assume to have a dataset of n samples $O_1, ..., O_n$, where $O_i \in \mathbb{R}^p$, and each sample is drawn i.i.d. from some distribution $O_i \sim P_O$. The goal of GANs is to use these samples to learn a model that can draw samples from the distribution P_O, usually a Gaussian or a uniform. At the same time, the discriminator takes samples as input and classify as either real or fake. Backpropagation is used to minimize the errors in the classification task to train the parameters of both the generator and discriminator. As soon as the training begins, the generator generates fake data, and the discriminator learns that the data is fake. After some training rounds, the generator learns how to fool the discriminator, resulting in the identification of fake data as real data. The loss function for GANs is: $min_G max_D \mathbb{E}_{x\sim p_x}[logD(x)] + \mathbb{E}_{z\sim p_z}[log(1 - D(G(z)))]$. In contrast to the prior generative modeling approach, e.g., likelihood maximization of parametric models, GANs make very few assumptions about the original data structure. The original GAN process [6] is presented in Fig. 1

GAN has been widely used in fake image generation and vanilla GAN have proven the success. The real-valued or continuous variables in the image data can be modeled using Gaussian distribution. Generating images of specific categories rather than high quality is still a challenging task as high variance is present in the specific image category. The authors in [20] states that when min-max normalizer is used on pixel values data to normalize, it is followed the Gaussian-like distribution. However, it is not always true for continuous variables in domains like computer vision that the distribution come from Gaussian. Moreover, min-max normalization can lead to vanishing gradient descent problem. This raises the concerns to handle non-image continuous data and other discrete types data.

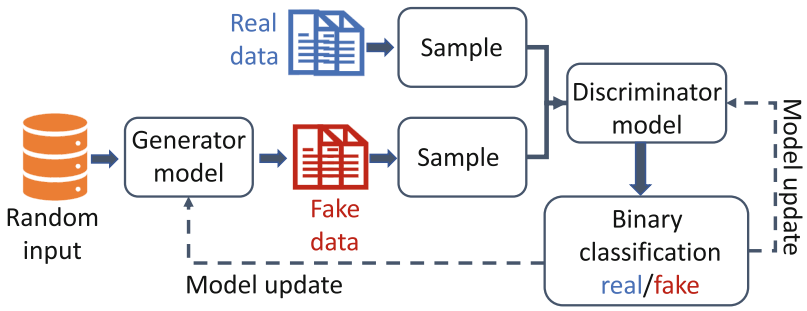

Fig. 1. GAN process

One of the well-known issue with the GAN is mode-collapse [6]. In mode-collapse, GAN generates a single type of output or small set of outputs despite being trained on diverse dataset. This issue happens when the generator finds a sample type that can fool the discriminator and it keeps generating that type as there is no incentive for the generator to change things up. Mode-collapse can cause serious issue when dealing with intrusion detection dataset as the dataset contains diverse types of data. Authors in the paper [13] claim to have dealt with the mode-collapse problem. Measurements of physical properties (metadata) can

also influence the characteristics of the measurements. For example, a denial-of-service (DoS) attack can create a larger traffic than a probing attack. So that GAN needs to learn the joint distribution between metadata and their corresponding measurements. The following steps were used in the DG paper to generate synthetic data:

- Mode-collapse
- Capturing attribute relationship

To tackle the mode collapse, DG develops a custom *auto-normalization* heuristics. For example, we have a dataset which has two time series with different offsets (min max values). A standard normalization approach would first normalize the data by the global min and max and store them as global constants. Then, GAN model train on the normalized data where normalization is just scaling and shifting by a constant. So the mode collapse still occurs. Instead DG, normalize each time series signal individually and store the min/max as fake metadata. Thus GAN learns to generate fake metadata by identifying the min.max for each time series individually, which are then used to rescale measurements to a realistic range. In our case, intrusion detection dataset, we consider each network flow as an individual time series signal. As in the intrusion detection dataset, each traffic flow at each time-step defined as malicious or benign traffic.

DG achieves in capturing the attribute relationship by introducing an auxiliary discriminator. Because, generating metadata and measurements using a single discriminator is too difficult. The auxiliary discriminator only discriminates on metadata. The losses from discriminator and auxiliary discriminator are then combined by a weighted parameter $\alpha : min_G max_{D_1,D_2} \mathbb{L}_1(G, D_1) + \alpha \mathbb{L}_2(G, D_2)$ where $\mathbb{L}_i, i \in \{1, 2\}$ is the Wasserstein loss of original and auxiliary discriminator respectively.

3.2 Applying Differential Privacy (DP)

User privacy is a concern in the digital industry with the growing use of data for a multitude of data-driven applications such as machine learning models. In the case of data sharing, privacy is the primary aspect to consider due to the disclosure of sensitive content to third parties. An existing method such as de-identification protects user privacy by selectively removing information fields connected to user identities. However, de-identification is prone to reconstruction attacks, i.e., queries forged on a database to reconstruct individual records. Further, de-identified databases are prone to linkage attacks where malicious actors can re-identify users by correlating the remaining fields with background information, i.e., using auxiliary data sources and forming a big picture of user profiles.

Differential privacy ensures users' privacy from reconstruction attacks by manipulating the output of an analytical query on a database. A differentially private algorithm guarantees that its outcome changes under controlled conditions (privacy budget), regardless of the data's single records (elements).

In the case of machine learning, DP states that a model $\mathcal{M}$ is differentially private if for any pairs of training datasets $\mathcal{D}$ and $\mathcal{D}^{'}$ that differ for a single user record, and for any input z, it holds that [13]

$$\mathcal{M}(z;\mathcal{D}) \leq e^{\epsilon}\mathcal{M}(z;\mathcal{D}^{'}) + \delta$$

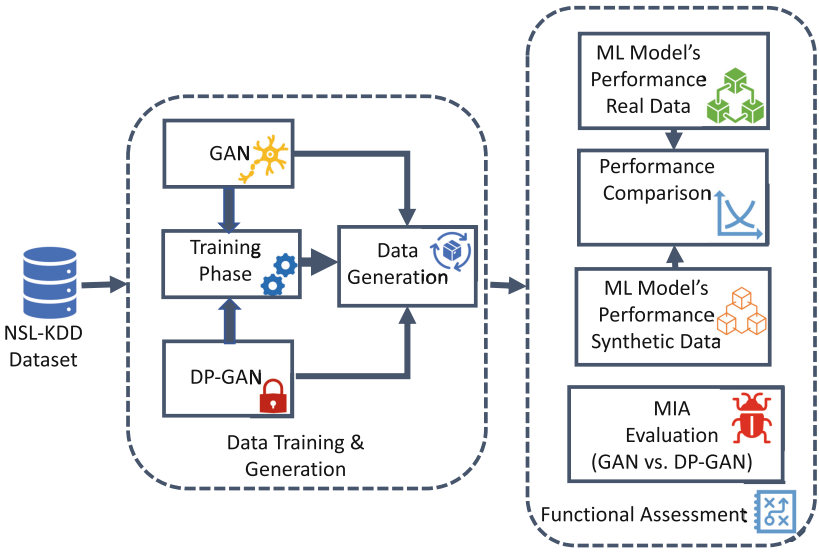

Fig. 2. Experimental flow diagram

where $\mathcal{M}(z;\mathcal{D})$ denotes a model trained on $\mathcal{D}$ and evaluated on z. Smaller values of ϵ and δ give more privacy. DoppleGANger's differential privacy framework is developed on top of the Google Tensorflow Privacy library [7]. The basic idea of Tensorflow Privacy is to modify the gradients used in stochastic gradient descent (SGD), which is the modified version of vanilla SGD. The first modification is done on the sensitivity of each gradient which is bounded by clipping the gradient computed for each training. The second modification is done with random noise sampled and added to the clipped gradients, hence making it impossible to identify which data point was used during training. The privacy optimizers share some privacy-specific parameters that need to be tuned before training the model, specifically [7]:

- *l2_norm_clip* : The maximum Euclidean (L2) norm of each individual gradient that is computed on an individual training example from minibatch.
- *noise_multiplier*: The amount of noise sampled and added to gradients during training the model.

Our experimental flow diagram is presented in Fig. 2. Initially, to understand the insights of the DG's GAN model where it captures the correlations between

measurements and their metadata, we conducted the same experiment described in the paper [13]. Then, in the data training & generation phase, we first train the GAN framework by feeding the NSL-KDD dataset and then train the DP-GAN framework. Finally, both (GAN and DP-GAN) synthetic data is generated.

4 Use Case Scenario: Data Sharing for Algorithm Training

4.1 Dataset Description

KDDCUP'99 is the most widely used dataset for intrusion detection. Despite the fame, the dataset suffers from two critical issues that highly affect the performance of intrusion detection systems trained on it, resulting in a limited validity. The first issue is the significant number of redundant records, causing biases on the learning algorithm towards those frequent records. Another issue is the level of complexity, as authors in [18] show that various evaluations as accuracy, detection rate, and false-positive rate are not a proper use of the KDD dataset. To solve these problems, authors in [18] proposed the NSL-KDD dataset, which consists of selected records of the complete KDD dataset.

The NSL-KDD dataset contains network traffic extracted from the real network environment consisting of normal (benign) traffic and malicious traffic. There are four main categories of attacks, namely Probing (Probe), Denial of Service (DoS), user to Root (U2R), and Root to Local (R2L). Extracted features from the raw traffic data are labeled as benign or malicious traffic. The 41 features consist of 9 discrete and 32 continuous values. Features are also categorized based on 4 semantic categories, including *intrinsic*, *content*, *time-based traffic*, and *host-based traffic*, as presented in Fig. 3. The dataset has 125 k records for a total size of 20 MB.

We use the Pearson Correlation Coefficient (PCC) for feature extraction in our experiment. PCC measures the strength and direction of a linear relationship between two variables. For our case, correlation coefficients whose magnitude is 0.5 indicate variables that can be considered moderately correlated. There are 9 features have been selected form NSL-KDD dataset which are, *count*, *srv_serror_rate*, *serror_rate*, *dst_host_serror_rate*, *dst_host_srv_serror_rate*, *logged_in*, *dst_host_same_srv_rate*, *dst_host_srv_count*, *same_srv_rate*.

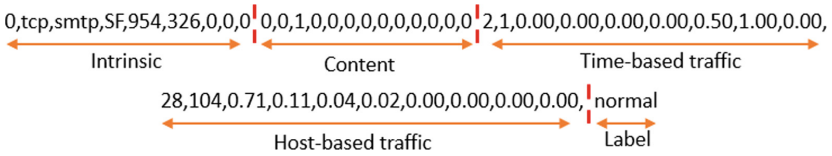

Fig. 3. Features description

Table 1. Non-identifier features examples

Feature type	Features examples
Categorical	protocol_type, service, flag, label
Continuous	src_bytes, dst_bytes, num_failed_login
Discrete	logged_in, root_shell, su_attempted

4.2 Privacy Concerns for the Dataset

IDS is one of the most effective defense mechanisms to protect from cyber-attacks if equipped adequately with updated rules and signatures. In particular, for machine learning-based IDS, the availability of high-quality and updated data is essential to deal with new versions of hacker tools. Data sharing offers a way to increase the knowledge base of security teams in a relatively short time and proactively. Privacy is, however, a key stumbling block for the research community because of contents not suitable to be shared. Privacy-sensitive data in IDSs can be found from three cases: IDS input data, IDS built-in data, and IDS generated data [15]. Privacy-sensitive fields in these cases can be present in two fields: privacy-preserving identifiers (i.e., user-names, IP address) and privacy-preserving non-identifiers (i.e., time-stamps, attack signature). In this paper, we focus on privacy-preserving non-identifiers, i.e., attack signatures.

IDS built-in data includes attack signatures to be used in misuse-based IDSs. Security vendors usually consider attack signatures as a piece of proprietary information to not be revealed to competitors. Furthermore, attackers can analyze signatures to learn potential vulnerabilities in target systems and design their exploits. Therefore, devices running an IDS can become an accessible source of information for attackers who want to learn and design new attacks. As we mentioned in the earlier section, there are four categories of attacks in the NSL-KDD dataset. All the fields in the dataset are non-identifiers, meaning there are no user names or IP addresses. Therefore, this paper aims to protect non-identifier information and generate a synthetic version of NSL-KDD while preserving the attack signature information. In Table 1, we show samples of non-identifier information from the dataset.

4.3 Data Preprocessing

Dataset needs to be processed before feeding into the training model. In our experiments we assess the performance with two synthetic data generation approaches as provided by DoppleGANger, a) intrusion detection accuracy without differential privacy, b) intrusion detection accuracy with differential privacy. Both approaches require us to pre-process the dataset.

Pre-Processing for ML: NSL-KDD dataset contains 41 features of multiple types and ranges. To apply the DG tools, we perform numeric conversion and

normalization. Among the 41 features, 9 are discrete and 32 are continuous. In the case of 9 discrete features, 3 features are non-numeric and 6 features binary (0,1). We use one-hot encoding to convert the non-numeric values to numeric values, such as "protocol_type" (TCP, UDP, ICMP).

Further, we use standard scalar to eliminate the dimensional impact between features values. For all discrete and continuous features, the min-max normalization method is used to convert the numeric values into the interval of [0,1]:

$$y^{'} = \frac{y - y_{min}}{y_{max} - y_{min}}$$

where, y represents the value before normalization for a specific feature in the dataset and $y^{'}$ is the value after the normalization.

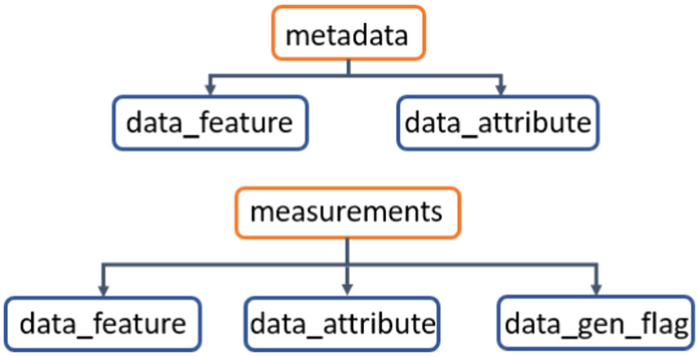

Fig. 4. Dataset formatting for DP-GAN

Pre-Processing for DP-GAN: To train the DP-GAN model, dataset needs to be formatted into the *metadata* and *measurements* categories [13] as show in Fig. 4. In the schema for *metadata*, *data_feature* and *data_attribute* represent a list of objects, indicating dimension, type, and normalization of each feature and attributes in the *Python Pickle format* (e.g., data_feature_output.pkl). The *data_attribute* in the *metadata* are *protocol_type*, *service*, *flag*. The remaining 38 features for dimension, type, and normalization are present in the *data_feature*.

In the schema for *measurements*, *data_feature* represents the value of different features in NumPy 3D array format. *data_attribute* represents the one-hot encoding of 3 categorical features (*protocol_type*, *service*, *flag*) and label of each record i.e. benign and malicious. *data_gen_flag* represents the time-series activation. All values are stored in an NPZ format with three arrays [13]:

– data_feature: Training features are stored in numpy float32 array format. The size is [(number of training samples) x (maximum length) x (total dimension of features)]. Categorical features are stored by one-hot encoding; for example,

Table 2. Accuracy in detection of normal and malicious traffic in original dataset

Classification algorithm	Training accuracy	Testing accuracy
Random Forest	99.42	98.69
Linear Support Vector Machine	97.84	97.8
Logistic Regression	96.99	96.99
Gaussian Naive Bayes	84.45	84.33
Gradient Boosting	98.24	98.21
Multi-Layer Perceptron	98.61	98.51

if a categorical feature has 3 possibilities, then it can take values between [1., 0., 0.], [0., 1., 0.], and [0., 0., 1.]

- data_attribute: Training attributes are stored in numpy float32 array format. The size is [(number of training samples) x (total dimension of attributes)]. Categorical attributes are stored by one-hot encoding; for example, if a categorical attribute has 3 possibilities, then it can take values between [1., 0., 0.], [0., 1., 0.], and [0., 0., 1.].
- data_gen_flag: Flags indicating the activation of features, in numpy float32 array format. The size is [(number of training samples) x (maximum length)]. 1 means the time series is activated at this time step, 0 means the time series is inactivated at this time step. In our case, data_gen_flag remains activated (1) all the time as the dataset does not have missing values.

5 Experimental Evaluation

We use Python (v3.7.10) and Tensorflow (v1.14.0) for the experiments. Tensorflow can be run on CPU or GPU; however, for our experiment, we use GPU_Task_Scheduler library, which is computationally faster than CPU. To record the intrusion detection accuracy, we tested 6 supervised machine learning models, specifically Random Forest (RF), Linear Support Vector Machine (LSVM), Logistic Regression (LR), Gaussian Naive Bayes (GNB), Gradient Boosting (GB), and Multi-Layer Perceptron (MLP).

To start with the normal and malicious traffic classification on the original NSL-KDD dataset, we process it as mentioned in Sect. 4. We adopt correlation-based feature selection to reduce dimensionality, thus obtaining a reduced set of 9 features. We divide the dataset in two, with 0.75 for training and 0.25 for testing. The accuracy in detecting normal and malicious traffic on the original dataset is presented in Table 2.

In the next phase of our experiment, we feed the processed dataset as described in Sect. 4. C to generate a pure GAN-based synthetic dataset. In Table 3, some of the core parameters of the GAN-based framework [13] are reported.

Table 3. GAN model parameter

Parameter	Value	Meaning
batch_size	100	Training batch size
d_rounds	1	Number of discriminator steps per batch
g_rounds	1	Number of generator steps per batch
g_lr	0.001	Learning rate in Adam for training the generator (1/s)
d_lr	0.001	Learning rate in Adam for training the discriminator (1/s)
attr_disc_num_layers	5	Number of layers in the auxiliary discriminator
attr_disc_num_units	200	Number of units in each layer of the auxiliary discriminator
disc_num_layers	5	Number of units in each layer of the auxiliary discriminator
initial_state	random	"random" means setting the initial state to random numbers
l2_norm_clip	1.0	Bound the optimizer's sensitivity to individual training points
noise_multiplier	[1.0, 2.0, 4.0]	Amount of noise sampled and added to gradients during training

Table 4. Accuracy in detection of normal and malicious traffic in synthetic dataset

Classification algorithm	Training accuracy	Testing accuracy
Random Forest	99.99	99.6
Linear Support Vector Machine	98.25	97.93
Logistic Regression	96.85	96.44
Gaussian Naive Bayes	90.71	90.61
Gradient Boosting	99.28	99.15
Multi-layer Perceptron	99.15	98.92

After the training, the GAN framework generates samples with a mix of normal and malicious traffic. The pure GAN generates 125K samples for training and 50K samples for testing. The accuracy detection on the synthetic dataset

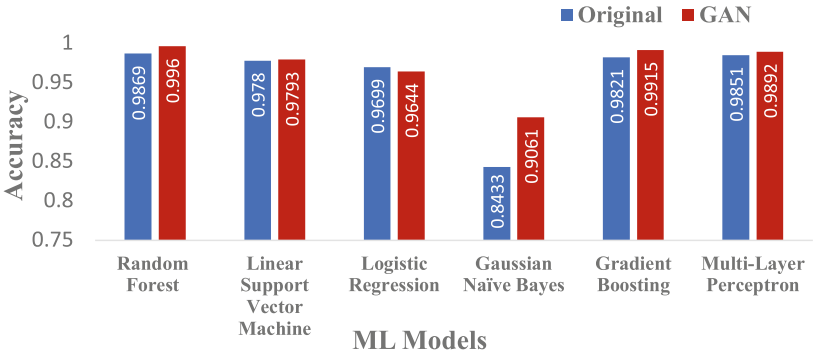

Fig. 5. Classification accuracy comparison (original vs GAN)

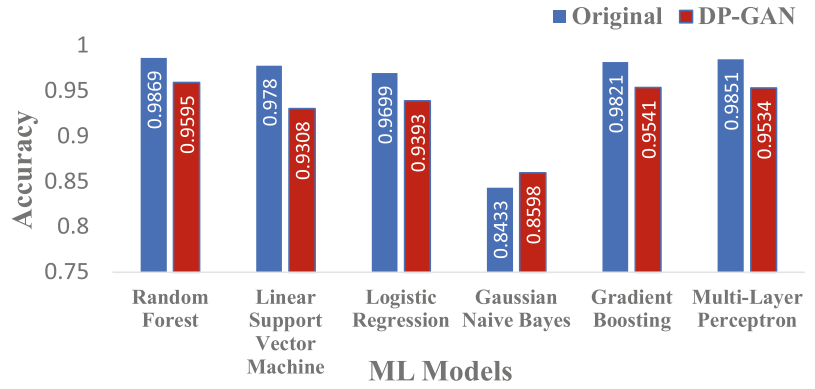

Fig. 6. Classification accuracy comparison (original vs DP-GAN)

is presented in Table 4. In Fig. 5, we present a comparison of testing accuracy between the original dataset and GAN-based synthetic data. It is notable from Fig. 5 that all the machine learning models perform well in classifying malicious traffic on the synthetic dataset than the original dataset. In the original NSL-KDD dataset, the distribution between malicious and normal traffic is imbalanced. On the other hand, the malicious and normal traffic distribution is balanced in the synthetic dataset. We observe that the GAN-based synthetic dataset gives high accuracy in classifying malicious traffic from normal traffic.

In the next phase of the experiments, we assess the accuracy of the differential privacy GAN (DP-GAN). We train the model using DoppleGANger's [13] differential privacy framework, which is based on the Google Tensorflow Privacy library [7]. The trained model generates synthetic samples for malicious and normal traffic. Machine learning models of the previous assessment are also applied to the differentially private synthetic dataset. Classification accuracy is presented in Fig. 6 for $\varepsilon = 6.73$. Here, we note that the detection accuracy for malicious traffic is close to the detection of malicious traffic on the original dataset. Since

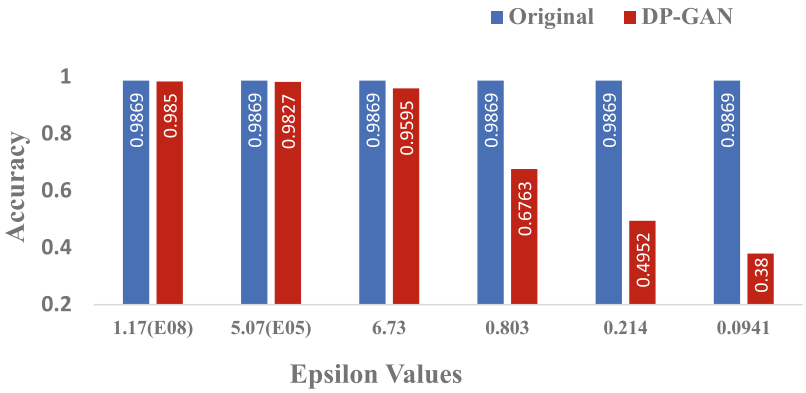

Fig. 7. Classification accuracy for different privacy budget values

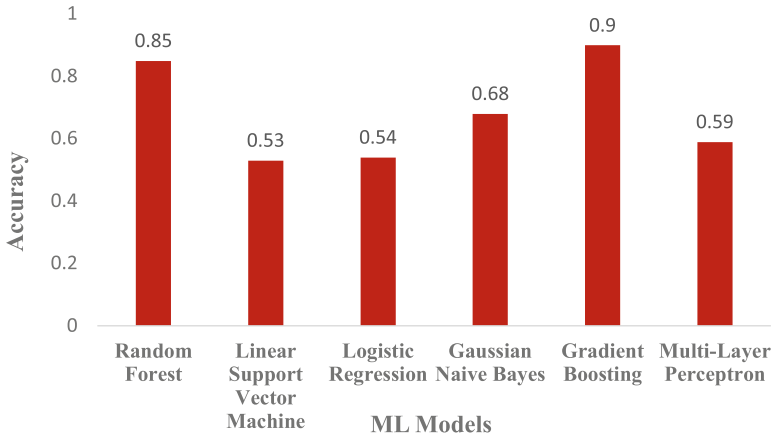

Fig. 8. Classification accuracy when ML models are trained on DP-GAN dataset and test on original dataset.

DP-GAN provides a model with a differential privacy budget, we can now assess the detection accuracy against several values of privacy parameters. In order to proceed in the assessment with DP, we record the accuracy for different ε values as depicted in Fig. 7 for Random Forest algorithm. The value on the x-axis represents different epsilon values (the letter E represents power of 10 as exponent). We observe that lowering the budget value deteriorates the accuracy to an unacceptable level. We also observe that when the privacy budget parameter is $\varepsilon = 6.73$, the accuracy is 95%, which is close to the original accuracy. For $\varepsilon = 6.73$, we obtain a good trade-off between differential privacy and accuracy. Finally, we assess the DP-GAN model in a realistic scenario, where it is trained on the syntehtic dataset and it is tested on the original dataset, while keeping the same privacy budget parameter found above. The generated dataset has 126k samples

Table 5. Confusion Matrix when models are trained on synthetic dataset and tested on original dataset

ML models	Accuracy	Precision	Recall	F1-score
RF	0.8456	0.9127	0.7863	0.8448
LSVM	0.5292	0.6969	0.2112	0.3242
LR	0.5440	0.7711	0.2091	0.3290
GNB	0.6831	0.6348	0.9586	0.7638
GB	0.9038	0.9060	0.9149	0.9104
MLP	0.5893	0.7835	0.3203	0.4547

for training the ML models. We observe Random Forest and Gradient Boosting models perform well on the original dataset where the accuracy is 85% and 90% respectively, as shown in Fig. 8. However, the other models seem to suffer from generative effects, thus showing worse performance than training with original data.

To further understand ML models' performance, we use three well-known metrics: Precision, Recall, and F1-Score. These metrics depend on four basic attributes, as follows:

- True Positive (TP) - Attack data which is correctly classified as an attack.
- True Negative (TN) - Benign data which is correctly classified as benign.
- False Positive (FP) - Benign data which is incorrectly classified as an attack.
- False Negative (FN) - Attack data which is incorrectly classified as benign.

The accuracy is the percentage of total correct classifications, where precision - i.e., TP/(TP+FP) - measures the number of positive classifications that belong to the positive class. Recall - i.e., TP/(TP+FN) - quantifies the number of correct class predictions made out of all positive examples in the dataset. F1-score - i.e., (2*Precision*Recall)/(Precision+Recall) - provides the harmonic mean of precision and recall in one number. From Table 5, it is evident that Random Forest and Gradient Boosting both models F1-score is close to the value of accuracy. Hence, we can conclude that both models (RF and GB) performed well.

Membership Inference Attacks Evaluation: GAN models are susceptible to Membership Inference Attacks (MIA). In the membership inference attacks, the attacker aims at inferring whether trained data samples have been used to train a machine learning model, hence revealing content in the original dataset. A simple MIA model is presented in Fig. 9. Authors in [4] argue that a smaller training dataset leads to a higher risk of revealing information used in training. This raises the concern when dealing with a real-world privacy-sensitive dataset (e.g., intrusion detection samples), and differential privacy is an effective defense mechanism against MIA on GAN training models [4].

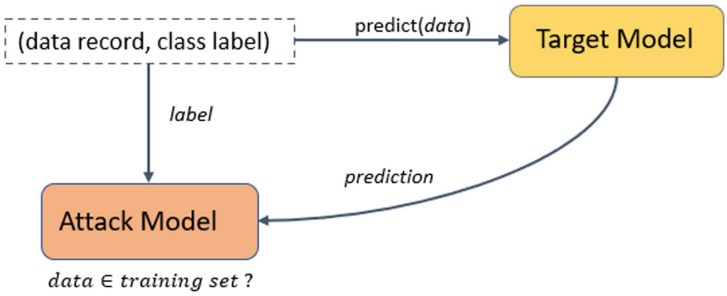

Fig. 9. Membership inference attack model

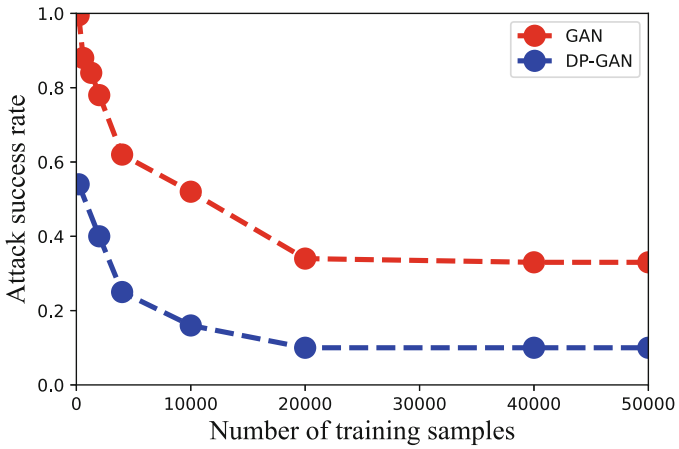

Fig. 10. Membership inference attacks against DP-GAN based dataset

To measure GAN model robustness against MIA, we use attack success rate [13] as a metric. We find that the attack success rate is high when the training sample size is small. For instance, with 200 training samples, the attack success rate is 0.998. However, increasing the sample size, the attack success rate decreases, as shown in Fig. 10. For the DP-GAN model, we note that, with 200 training samples, the attack success rate is 0.54, which is lower than GAN. Our results suggest that the differential privacy-based GAN model can effectively defend against MIA using more training data as compared to pure GAN models.

6 Conclusion

In this paper, we investigated methods to protect privacy in the case of data sharing and for training machine learning algorithms for intrusion detection. First, we assessed the quality of synthetic datasets generated with Generative Adversarial Networks (GANs) and Differential Privacy, particularly with the

DoppleGANger toolset. We assessed the quality of synthetic data in terms of detection accuracy, i.e., ability to classify malicious vs. benign network traffic. We use the well-known intrusion detection dataset NSL-KDD. The experimental results showed that the synthetic dataset with differential privacy (DP-GAN) could achieve high classification accuracy (95.95%) while maintaining a low privacy budget parameter ($\varepsilon = 6.73$), i.e., the low success rate for member inference attacks. We also observed a 90% accuracy when the model was trained on the DP-GAN dataset and tested on the original dataset. Our results suggested a practical guideline: dataset owners can generate differentially private synthetic datasets and share the dataset among researchers and cyber-security practitioners without privacy concerns. Then, the researcher can develop ML models to achieve high binary (malicious or benign) and multiclass (attack categories, i.e., probe, DoD, R2L, U2R) classification accuracy.

References

1. Abadi, M., et al.: Deep learning with differential privacy. In: Proceedings of the 2016 ACM SIGSAC conference on computer and communications security, pp. 308–318 (2016)
2. Brock, A., Donahue, J., Simonyan, K.: Large scale gan training for high fidelity natural image synthesis. arXiv preprint arXiv:1809.11096 (2018)
3. Chakrabarti, S., Chakraborty, M., Mukhopadhyay, I.: Study of snort-based ids. In: Proceedings of the International Conference and Workshop on Emerging Trends in Technology, pp. 43–47 (2010)
4. Chen, D., Yu, N., Zhang, Y., Fritz, M.: Gan-leaks: A taxonomy of membership inference attacks against generative models. In: Proceedings of the 2020 ACM SIGSAC conference on computer and communications security, pp. 343–362 (2020)
5. Dandekar, A., Zen, R.A.M., Bressan, S.: A Comparative Study of Synthetic Dataset Generation Techniques. In: Hartmann, S., Ma, H., Hameurlain, A., Pernul, G., Wagner, R.R. (eds.) DEXA 2018. LNCS, vol. 11030, pp. 387–395. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-98812-2_35
6. Goodfellow, I., et al.: Generative adversarial nets. In: Advances in Neural Information Processing Systems, vol. 27 (2014)
7. Google: Tensorflow privacy. https://github.com/tensorflow/privacy
8. Huang, S., Lei, K.: Igan-ids: an imbalanced generative adversarial network towards intrusion detection system in ad-hoc networks. Ad Hoc Netw. **105**, 102177 (2020)
9. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1125–1134 (2017)
10. Jordon, J., Yoon, J., Van Der Schaar, M.: Pate-gan: Generating synthetic data with differential privacy guarantees. In: International conference on learning representations (2018)
11. Kotal, A., Piplai, A., Chukkapalli, S.S.L., Joshi, A., et al.: Privetab: Secure and privacy-preserving sharing of tabular data. In: ACM International Workshop on Security and Privacy Analytics (2022)
12. Lin, Z., Shi, Y., Xue, Z.: Idsgan: Generative adversarial networks for attack generation against intrusion detection. arXiv preprint arXiv:1809.02077 (2018)

13. Lin, Z., Jain, A., Wang, C., Fanti, G., Sekar, V.: Using gans for sharing networked time series data: Challenges, initial promise, and open questions. In: Proceedings of the ACM Internet Measurement Conference, pp. 464–483 (2020)
14. Mukherjee, S., Sharma, N.: Intrusion detection using naive bayes classifier with feature reduction. Procedia Technol. **4**, 119–128 (2012)
15. Niksefat, S., Kaghazgaran, P., Sadeghiyan, B.: Privacy issues in intrusion detection systems: a taxonomy, survey and future directions. Comput. Sci. Rev. **25**, 69–78 (2017)
16. Salem, M., Taheri, S., Yuan, J.S.: Anomaly generation using generative adversarial networks in host-based intrusion detection. In: 2018 9th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), pp. 683–687. IEEE (2018)
17. Shahriar, M.H., Haque, N.I., Rahman, M.A., Alonso, M.: G-ids: Generative adversarial networks assisted intrusion detection system. In: 2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC), pp. 376–385. IEEE (2020)
18. Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.A.: A detailed analysis of the kdd cup 99 data set. In: 2009 IEEE symposium on computational intelligence for security and defense applications, pp. 1–6. IEEE (2009)
19. Xie, L., Lin, K., Wang, S., Wang, F., Zhou, J.: Differentially private generative adversarial network. arXiv preprint arXiv:1802.06739 (2018)
20. Xu, L., Skoularidou, M., Cuesta-Infante, A., Veeramachaneni, K.: Modeling tabular data using conditional gan. In: Advances in Neural Information Processing Systems 32 (2019)
21. Yu, Z., Tsai, J.J.: A framework of machine learning based intrusion detection for wireless sensor networks. In: 2008 IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (sutc 2008), pp. 272–279. IEEE (2008)
22. Zhang, L., Jiang, S., Shen, X., Gupta, B.B., Tian, Z.: Pwg-ids: An intrusion detection model for solving class imbalance in iiot networks using generative adversarial networks. arXiv preprint arXiv:2110.03445 (2021)