Improving Intrusion Detection for Imbalanced Network Traffic using Generative Deep Learning

Amani A. Alqarni
Department of Computer Science and Engineering
College of Computer Science and Engineering
University of Hafr Al Batin
Hafr Al Batin 39524, Saudi Arabia

El-Sayed M. El-Alfy
Information and Computer Science Department
Interdisciplinary Research Center for Intelligent Secure Systems
College of Computing and Mathematics
King Fahd University of Petroleum and Minerals
Dhahran 31261, Saudi Arabia

***Abstract*—Network security has become a serious issue since networks are vulnerable and subject to increasing intrusive activities. Therefore, network intrusion detection systems (IDSs) are an essential component to defend against these activities. One of the biggest issues encountered by IDSs is the class imbalance problem which leads to a biased performance by most machine learning models to normal activities (majority class). Several techniques were proposed to overcome the class-imbalance problem such as resampling, cost-sensitive, and ensemble learning techniques. Other issues related to intrusion detection data include mixed data types, and non-Gaussian and multimodal distributions. In this study, we employed a conditional tabular generative adversarial network (CTGAN) model with common machine learning algorithms to construct more effective detection systems while addressing the imbalance issue. CTGAN can generate samples of the minority class during training to make the dataset more balanced. To assess the effectiveness of the proposed IDS, we combined CTGAN with three machine learning algorithms: support vector machine (SVM), K-nearest neighbor (KNN), and decision tree (DT). The imbalanced NSL-KDD dataset was used and several experiments were conducted. The results showed that CTGAN can improve the performance of imbalance learning for intrusion detection with SVM and DT. On the other hand, KNN showed no improvement in the performance since it is less sensitive to the class imbalance problem. Moreover, the results proved that CTGAN can capture the distribution of discrete features better than continuous features.**

Keywords*—Intrusion detection; machine learning; imbalance learning; conditional tabular generative adversarial networks

I. Introduction

In the current era of transformation to digital services and with the evolution of Internet technologies, cybersecurity has become a serious issue, especially with the growing volume and diversity of attacks. People utilize the Internet to conduct most of their work (e.g. online shopping, payment, banking, access to governmental services, file sharing, communication, and more). Moreover, it is a vital part of several cyber-physical systems in critical infrastructures such as Internet-of-Things (IoT) and smart grids. Therefore, intrusion detection systems are an essential component of cybersecurity and provide crucial services to protect information systems from cyberattacks that can lead to catastrophic consequences (e.g. sensitive data leakage, physical harm, and financial loss). It can monitor the system operations and network traffic to detect anomalous patterns [1].

Intrusion detection is a recurrent research topic due to the emergence of more sophisticated adversarial incidents. One of the challenges facing intrusion detection is the scarcity of intrusive samples compared to the abundance of normal operation samples; resulting in insufficient data samples to train the model on a representative collection of attack scenarios. With such an imbalanced dataset, the number of samples in a majority (normal or negative) class is significantly higher than the number of samples in a minority (abnormal or positive) class. Training a machine learning algorithm on an imbalanced dataset can adversely impact the performance and lead to unsatisfactory results since it can be more biased towards the majority class, i.e. normal patterns.

Imbalanced learning is very common in several problems with rare events and different techniques have been proposed to address the imbalance issue. These techniques can be divided into five main categories: data-level, algorithm-level, cost-sensitive, ensemble-based, and hybrid techniques. The data level are preprocessing techniques known as re-sampling because they either increase the frequency of the minority class (oversampling) or reduce the frequency of the majority class (undersampling/downsampling). Random oversampling (ROS), random undersampling (RUS), synthetic minority oversampling technique (SMOTE), and generative adversarial networks (GAN) are examples of data-level techniques. A systematic literature review of the challenges and solutions for imbalanced data in machine learning is provided in [2].

The focus of our study is on oversampling techniques, specifically, an enhanced version of GAN, i.e. the conditional tabular GAN (CTGAN) [3]. CTGAN is a recent deep learning model and can be thought of as an oversampling technique. It can augment the tabular dataset and increase the frequency of the minority class samples while handling other issues such as mixed data types, multimodality, and non-Gaussian distributions. It proves its efficiency in addressing the imbalance problem and improving the classification accuracy in different domains.

This paper aims to investigate the role of CTGAN in improving the classification performance of support vector machines (SVM), K-nearest neighbors (KNN), and decision trees when applied to imbalanced data to detect various types of network intrusions. Different metrics have been computed to evaluate the quality of the generated data for various attacks in the multi-class NSL-KDD dataset. Moreover, the performance measures of the trained intrusion detection models have been computed and discussed.

The main contributions of this paper are:

- Handling the imbalance problem in intrusion detection datasets by employing CTGAN which is not investigated very well in the literature.
- Evaluating the quality of the generated data by CTGAN in terms of different metrics.

The rest of this paper is organized as follows: Section II provides a brief background about intrusion detection, class imbalance problem, and techniques to deal with this problem. Section III reviews related studies on intrusion detection. Section IV describes the methodology we followed in our study. Finally, Section V describes the experiments we conducted to evaluate the proposed prototype.

II. Background

A. Intrusion Detection

An intrusion detection system is an essential component responsible for analyzing and monitoring networks to detect intrusions and alert administrators on ongoing attack activities [4]. Intrusion detection is still a significant research field for two reasons. First, there are continuous updates and changes of network intrusions resulting in continually changing patterns [5]. Second, the number of available intrusion detection datasets is increasing over time, making it possible to investigate and compare new approaches [5]. Examples of intrusion categories include Denial of Service (DoS) such as smurf, User to Root (U2R) such as buffer overflow, Probing (Prob) such as portsweep, and Root to Local (R2L) such as password guessing [6]. An effective intrusion detection system should have not only low false negative but also low false positive. These measures can be greatly affected by the quality of having a representative training dataset. However, real scenarios may have several challenges. For example, besides the class imbalance problem, the intrusion detection traces may have several other characteristics that need a special treatment, e.g. mixed data types (continuous, discrete, ordinal, and categorical) as well as non-Gaussian and multimodal distributions.

B. Class Imbalance Problem

Most of the intrusion detection datasets are imbalanced datasets, which causes a degradation in the classification performance for certain types of intrusions [7]. Usually, the number of normal traces in intrusion detection datasets is much higher than the number of intrusion traces. Thus, as a minority class, the intrusion class might not be well-represented and hence not classified correctly.

The misclassification of the minority class costs more than the misclassification of the majority class, as it could cause a serious problem. Misclassifying a normal behavior class can lead to the need for more tests to explore the intrusion. On the other hand, misclassifying intrusions may lead to disaster impacts (e.g., privacy loss, unauthorized access to network assets, or damage of the whole system). Even if the detection rate of the minority class is low, classification accuracy could be high because the classification accuracy does not consider the distribution of classes. Consequently, machine-learning based intrusion detection systems can be accuracy biased, since they can give more attention to the majority class (a.k.a. normal behavior).

C. Imbalance Techniques

Many different techniques can be used to address the imbalance issue. These can be categorized into five categories, i.e. data-level, algorithm-level, cost-sensitive, ensemble-based, and hybrid techniques. Data-level techniques modify the class distributions before the training process. This modification is done either by removing some instances from the majority class or by adding more instances to the minority class [8]. The former method is known as undersampling while the latter is known as oversampling. Random oversampling (ROS), random undersampling (RUS) and synthetic minority oversampling technique (SMOTE) are examples of data-level preprocessing techniques.

Unlike data-level techniques, algorithm-level techniques do not modify the distribution of classes; but rather, they modify the algorithm [9]. In contrast, cost-sensitive techniques assign different costs to give the minority class higher importance than the majority class [10]. Ensemble methods combine more than one algorithm to achieve superior performance than would normally be attained separately such as bagging, boosting, stacking, and cascading classifiers [10]. Moreover, hybrid techniques combine two or more of the aforementioned techniques to produce an efficient technique for handling imbalance [9]. For instance, SMOTEENN combines oversampling by SMOTE with undersampling by the edited nearest neighbor (ENN) method [11].

D. Conditional Tabular GAN

Generative Adversarial Network (GAN) is one of the top innovative deep learning models [12]. It has been widely used in various applications to process different types of data (e.g., images, voice, and text). Hence, it became one of the most critical research fields in deep learning. It combines two networks: generator and discriminator [13]. The generator is responsible for generating synthetic data that resemble to the original data whereas the discriminator is responsible for classifying the real or synthetic data with their corresponding classes [13]. GAN in tabular data has various challenges. One of the challenges is that the structured data can follow non-Gaussian and multimodal distributions. Tabular GAN (TGAN) resolved this issue by using mode-specific normalization [3].

In GAN, the generator does not consider the imbalance issue. Data that belong to the minority class will not be presented sufficiently as the data belong to the majority class do. Conditional generator in conditional GAN can be used to enforce the synthetic sample to match a specific class (category) [14]. Hence, it can generate more intrusive samples to overcome the imbalance issue in intrusion detection datasets. To use the conditional generator, in CGAN, instead of the original generator in GAN, three issues must be mitigated:

- A way must be found to represent the condition and the input to the generator.
- The generated samples must match the chosen category (condition).

- The conditional generator should learn the conditional distribution.

Conditional Tabular GAN (CTGAN) combines the advantages of CGAN and TGAN. Therefore, it can be used to solve the class imbalance issue by controlling the class labels of the generated samples. It also overcomes the non-Gaussian and multimodal distributions of structured data. Furthermore, CTGAN utilizes fully connected networks to enhance the quality of the model [3]. The conditional generator can be interpreted as

$$\hat{\mathbf{r}} \sim P_g(row|D_{i*} = k^*)$$

where k^* is the chosen category from the discrete column D_{i*} that must be generated by conditional generator and $\hat{\mathbf{r}}$ is the sample generated by the generator. Fig. 1 shows a typical architecture of the CTGAN.

III. Related Work

With the growth of the amount of data related to intrusion detection, and the evolution in machine learning and deep learning techniques, many studies have been conducted in this field. Most of the previous studies ignore the class imbalance problem and use datasets of balanced distributions. In [1], the authors proposed a framework called scale-hybrid-IDS-AlertNet which can trace the network traffic and detect abnormal activities. The building of this framework came after a comprehensive analysis of different machine learning and deep learning models on different intrusion detection datasets. They found that the deep neural network (DNN) model outperforms other machine learning models such as Logistic Regression (LR), Naïve Bayes (NB), K-nearest neighbor (KNN), and support vector machines (SVM).

There are some researchers who have used imbalance techniques to deal with this problem in network intrusion detection systems. For example, Razan Abdulhammed et al. [15] compared the performance of different data-level techniques on the CIDDS-001 dataset. For data preprocessing, they considered ROS, RUS, class balanced and spread subsample. For classification, they utilized deep neural networks (DNN), random forest (RF), voting technique, stacking technique and variational autoencoder (VA). The superior performance was achieved by RF on the original distribution, class balancer and RUS (99.9%). Moreover, the accuracy of voting in the original distribution and using RUS was high (99.99%).

SMOTE is a known and effective oversampling technique, and many studies have been conducted to prove its efficiency. In [16], the imbalance issue was mitigated in a CICIDS2017 dataset using SMOTE oversampling. As SMOTE works only with binary classification, the researchers examined two classes at a time—a normal class with one of the minority classes (i.e., botnet, web attack, or brute force attack). Two experiments were conducted: one on the imbalanced dataset and one on the balanced dataset. Three algorithms were utilized to conduct the experiments (i.e., RF, NB, and KNN). On the imbalanced dataset, the accuracy was high with all classifiers, but the precision, recall, and F1-score were low. To mitigate this degradation, the researchers applied the SMOTE technique and the result showed better performance based on F1-score and recall.

Generative adversarial networks (GAN) is considered as a data-level technique because it modifies the distribution of data by generating new samples. It was applied in [17] by Yilmaz et al. to improve the performance of intrusion detection. The result after applying GAN was found to be more accurate than without using GAN. Different versions of GAN were proposed to enhance its performance. For example, Shuokang Huang and Kai Lei. [18] proposed Imbalanced GAN (IGAN) that includes a data imbalance filter, a generator, and a discriminator. It can force the generator to generate samples of the minority class only. IGAN with fuzzy neural network (FNN) obtained a superior performance over Convolutional Neural Network (CNN), RUS with SVM, FNN, and SMOTE with multilayer perception (MLP).

Punam Bedi et al. [19] proposed an intrusion detection system called Siam IDS which is based on Siamese Neural Network (Siamese-NN). It can handle the imbalance issue in intrusion detection systems. It achieved high recall values of the minority classes (U2R and R2L intrusions). Moreover, it outperformed CNN-based IDS and DNN-based IDS. In [20], m-RIGFS and RWIGFS were used with weighted-SVM to improve the imbalance learning for the intrusion detection systems. m-RIGFS and RWIGFS are feature selection techniques for imbalanced classes. This approach obtained a good performance in terms of the overall accuracy, sensitivity, and specificity. However, it should be noted that the sensitivity of the U2R, a rare class was low.

In [7], a CNN model was utilized to classify UNSW-NB15 and CICIDS2017, which are relatively recent intrusion detection datasets. They applied the CNN model, after addressing the class imbalance issue using their proposed approach, i.e. SGM which combines an oversampling technique (SMOTE) with an undersampling technique (Gaussian Mixture (GMM)). The proposed approach achieved a higher result (more than 96%) compared to other sampling techniques and classification models. Furthermore, in [21], SMOTE was combined with a genetic fuzzy system that includes a fitness function designed to deal with the imbalance problem. The proposed approach outperformed other approaches, which are the KDDCup-99 winner [22], GFS(Pittsburgh) [23], MOG-FIDS [24], EFRID [25] and RIPPER [26].

Data-level and cost-sensitive techniques can be combined to yield better performance. Alabdallah et al. in [27] combined a stratified sampling with a cost function. This approach assigns the minority class samples higher weights than the majority class samples to improve the classification performance and decrease the accuracy paradox.

To our knowledge and based on exploring earlier work in the literature, limited studies have used GAN-based methods to overcome the imbalance problem in intrusion detection systems. Moreover, none of the reviewed studies have evaluated the generated samples. Therefore, in our study, we explore CTGAN machine-learning based models to deal with the imbalance learning for intrusion detection. Moreover, we evaluated the generated samples in terms of different metrics (e.g. Chi-Squared test and Continuous Kullback–Leibler Divergence).

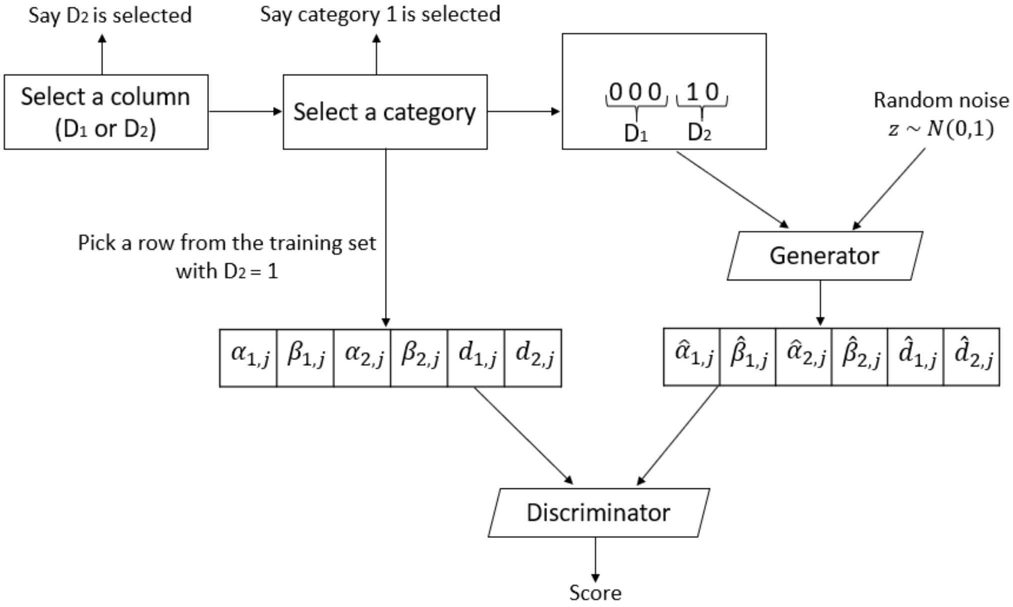
Say D2 is selected
Say category 1 is selected
Select a column (D1 or D2)
Select a category
000 10
D1 D2
Random noise
$z \sim N(0,1)$
Pick a row from the training set with D2 = 1
Generator
$\alpha_{1,j}$ $\beta_{1,j}$ $\alpha_{2,j}$ $\beta_{2,j}$ $d_{1,j}$ $d_{2,j}$
$\hat{\alpha}_{1,j}$ $\hat{\beta}_{1,j}$ $\hat{\alpha}_{2,j}$ $\hat{\beta}_{2,j}$ $\hat{d}_{1,j}$ $\hat{d}_{2,j}$
Discriminator
Score

Fig. 1. CTGAN Architecture.

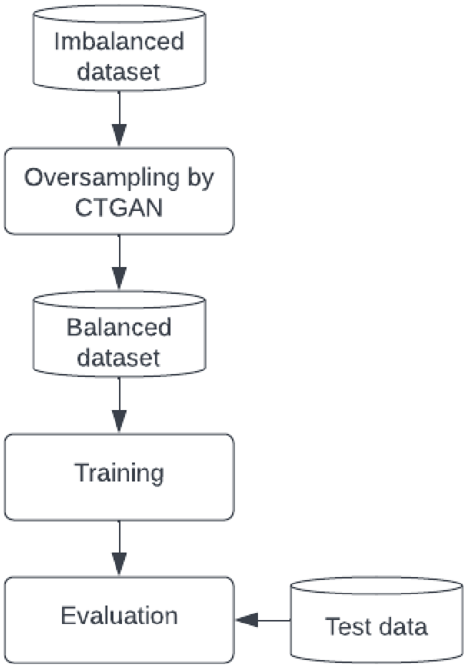

Fig. 2. General Layout of the Workflow.

IV. Methodology

Fig. 2 depicts the general layout of the workflow. The aim of this work is to investigate the performance of CTGAN in improving the detection rate of intrusive samples in intrusion detection systems. To achieve this aim, we applied the following steps:

- Oversampling by CTGAN: Since the intrusion data suffers from several issues including class imbalance, mixed data types, multimodality, and non-Gaussian distributions, we first applied CTGAN to increase the intrusive samples in the training dataset.
- Data preprocessing: The aim of this step is converting raw data into a suitable format for machine learning models. In this study, two preprocessing techniques were applied to the dataset before classification:
  - One-hot encoding: Some machine learning algorithms work only with numeric data, hence one-hot encoding is important when dealing with a dataset that contains categorical features. One-hot encoding is a method of sorting each categorical value into a distinct column and setting a value to it (either 0 or 1). Therefore, one-hot encoding was applied to the NSL-KDD dataset to convert its categorical features (i.e., protocol type, service, and flag).
  - Standard scalar: This is a significant step for some machine learning algorithms because features that are not scaled to have zero mean and unit variance could negatively affect the performance of the algorithm. This method scales the features to have a mean of zero and a standard deviation of one. Hence, all values will be in the same range. The following equation is used to perform standard scalar:

    $$z = \frac{x - \mu}{\sigma}$$

    where x is a sample, μ is the mean of the samples and σ is the standard deviation of the samples.
- Machine learning training and testing: To assess the efficiency of using CTGAN to overcome the imbal-

ance problem and other data issues, three machine learning algorithms (i.e. SVM, DT, and KNN) were used to train and test the NSL-KDD dataset before and after applying CTGAN.

- Evaluation: To accurately evaluate the performance of the proposed framework, various metrics (e.g. F1-score, geometric mean (G-mean), and Matthews correlation coefficient (MCC)) were used. These metrics are appropriate for evaluating the performance of imbalance learning.

V. Experiments and Evaluation

A. Dataset Description

NSL-KDD is an imbalanced dataset with huge amount of captured traffic under various normal and attack scenarios. It is an improved and revised version of the KDD99 dataset. It contains 41 features extracted from traffic traces of normal and abnormal activities as shown in Table I. Intrusion traffic is categorized into four main categories: Denial-of-Service (DoS), Probing (Probe), Remote-to-Local (R2L), and User-to-Root (U2R). Each category of these attacks contains several sub-categories as shown in Table II. In this study, only 20% of the dataset was used to conduct the experiments. Each class includes a different number of samples as shown in Table III. Therefore, this dataset is imbalanced because of the obvious difference in the number of samples in each class. The imbalance ratio of normal and attack classes in the dataset ranges from 1.44 to 305.77. Fig. 3 shows the distribution of intrusive and normal samples.

TABLE I. Features of NSL-KDD Dataset

No.	Feature	No.	Feature	No.	Feature
1	duration	16	num_root	31	srv_diff_host_rate
2	protocol_type	17	num_file_creations	32	dst_host_count
3	service	18	num_shells	33	dst_host_srv_count
4	flag	19	num_access_files	34	dst_host_same_srv_rate
5	src_bytes	20	num_outbound_cmds	35	dst_host_diff_srv_rate
6	dst_bytes	21	is_host_login	36	dst_host_same_src_port_rate
7	land	22	is_guest_login	37	dst_host_srv_diff_host_rate
8	wrong_fragment	23	Count	38	dst_host_serror_rate
9	urgent	24	srv_count	39	dst_host_srv_serror_rate
10	hot	25	serror_rate	40	dst_host_rerror_rate
11	num_failed_logins	26	srv_serror_rate	41	dst_host_srv_rerror_rate
12	logged_in	27	rerror_rate		
13	num_compromised	28	srv_rerror_rate		
14	root_shell	29	same_srv_rate		
15	su_attempted	30	diff_srv_rate		

TABLE II. Attack Categories in NSL-KDD

Attack category	Attack Types
DoS	Back, Apache2, Smurf, Neptune, Udpstorm, Land,Worm, Pod, Processtable, Teardrop
Prob	Satan, Portsweep, Nmap, Ipsweep, Mscan, Saint
R2L	Warezmaster, Phf, Ftp_write, Named, Snmpguess,Httptunnel, Xlock, Spy, Xsnoop, Sendmail, Imap,Guess_Password, Warezclient, Multihop, Snmpgetattack
U2R	Loadmodule, Buffer_overflow, Rootkit, Xterm,Sqlattack, Perl, PS

B. Performance Measures

1) Evaluation Metrics of Machine Learning Models: The following metrics were used to evaluate the performance of the applied classifiers:

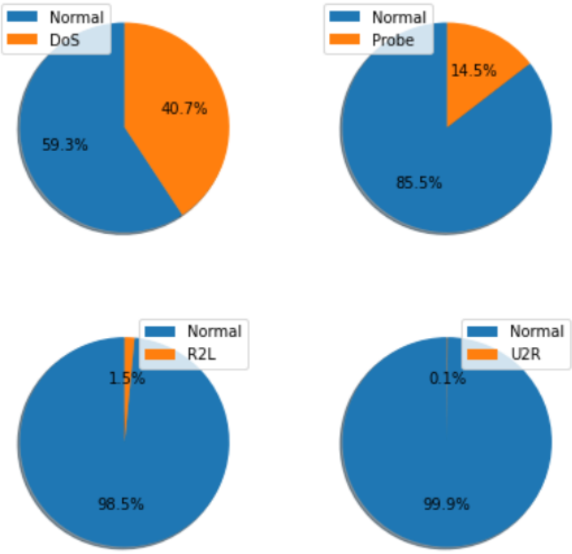

Fig. 3. Distribution of the Samples.

TABLE III. Samples in NSL-KDD

Class	Train set	Test set
Normal	13449	9711
DoS	9234	7458
Prob	2289	2421
R2L	209	2754
U2R	11	200
Total	25192	22544

a) Accuracy (ACC): This is the most common metric to evaluate the performance of a model. It is the number of samples that are correctly predicted over the number of all samples. It can be calculated using the following equation:

$$ACC = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

b) Recall: It refers to the ability of the model to predict positive samples. It can be calculated by dividing the number of the samples that are correctly classified as true positive over all positive samples. It can be calculated using the following equation:

$$Recall = \frac{TP}{TP+FN} \quad (2)$$

c) Precision: It is the number of samples that are correctly classified as true positive over the number of samples that are predicted as positive. The following equation can be used to calculate the precision:

$$Precision = \frac{TP}{TP+FP} \quad (3)$$

d) F1-score: It is a way to combine recall and precision into a single metric. It is called the harmonic mean of recall and precision. It can be calculated using Equation 4:

$$F1\_score = \frac{2 \times Precision \times Recall}{Precision+Recall} \quad (4)$$

e) Geometric mean (G-mean): It is based on the true positive rate (sensitivity or recall) and true negative rate (specificity). G-mean is a combination of sensitivity (recall), and specificity. Specificity and G-mean can be calculated using the following equations:

$$Specificity = \frac{TN}{FP+TN} \qquad (5)$$

$$G\_mean = \sqrt{Sensitivity \times Specificity} \qquad (6)$$

f) Matthews correlation coefficient (MCC): It is a good performance metric for binary classification and combines all parts of the confusion matrix (i.e., true positives, false positives, true negatives, and false negatives). It can be calculated using the following equation:

$$MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP+FP) \times (TP+FN) \times (TN+FP) \times (TN+FN)}} \qquad (7)$$

C. Synthetic Data Evaluation Metrics

The overall evaluation score of the synthetic data is an aggregation of the following metrics:

a) Chi-Squared (CSTest): It is a statistical metric that compares the distributions of two discrete columns using the Chi-squared test.

b) Inverted Kolmogorov-Smirnov D statistic (KSTest): It is a statistical metric that compares the distributions of the continuous columns using the Kolmogorov–Smirnov test.

c) KSTestExtended: It is an extension of the KSTest metric that transforms all columns into numerical columns before applying the KSTest.

d) Continuous Kullback–Leibler Divergence (Continuous KLDivergence): This metric calculates the Kullback-Leibler (KL) divergence on all pairs of the numerical columns.

e) Discrete Kullback–Leibler Divergence (Discrete KLDivergence): This metric calculates the Kullback-Leibler divergence on all pairs of the Boolean and categorical columns.

D. Experiments

In this study, one-vs-one classification was performed. We divided the dataset into four subsets. Each subset contains two class labels, i.e. normal and one of the intrusions (i.e., DoS, Prob, R2L, or U2R).

The experiments were conducted in Python and CTGAN was used to oversample the subsets using different number of epochs. Tuning epoch size in deep learning models is important, as it has a direct effect on the performance. It is the number of iterations the model performs over the training dataset. The default epoch size value in CTGAN is 300. In this study, different epoch size values (e.g., 300, 400, and 1600) were tested to obtain the best results.

Then, we evaluated the quality of the synthetic samples using multiple evaluation metrics that are combined to produce an overall score. This score provides a general indication of how good the synthetic samples are. Lastly, SVM, KNN, and decision tree models were run on the NSL-KDD dataset before and after implementing CTGAN.

E. Result and Discussion

Table IV presents the evaluation scores of the synthetic data used to balance the subsets. The overall evaluation score is the average of multiple scores that evaluate the data from different aspects (e.g., statistical, detection, and likelihood). The overall score gives an estimation of how similar the synthetic data and the real data are (i.e., the quality of the generated data). This score ranges from 0 to 1, where 0 is the worst possible score and 1 is the best possible score. As shown in Table IV all synthetic data of all attack categories achieved overall scores around 0.5.

We observed that CSTest obtained high scores ranging from 0.85 to 0.99, whereas KSTest achieved fairly good scores ranging from 0.77 to 0.82. Moreover, KL Divergence scores for discrete columns are reasonable unless for DoS; it is slightly high. On the other hand, KL Divergence scores for Continuous columns are high which indicates worse performance. Therefore, we can conclude that CTGAN can fairly capture distributions of both continuous and discrete columns, but its performance is better in the discrete columns.

TABLE IV. Evaluation Scores of the Synthetic Samples

Metric	DoS	Probe	R2L	U2R
Overall evaluation score	0.54	0.50	0.49	0.51
CSTest	0.99	0.98	0.85	0.92
KSTest	0.80	0.77	0.82	0.80
KSTestExtended	0.79	0.76	0.81	0.80
ContinuousKLDivergence	0.83	0.73	0.80	0.84
DiscreteKLDivergence	0.40	0.29	0.28	0.21

TABLE V. Results of one-vs-one SVM Classification

Imbalanced dataset	Normal-vs-DoS	Normal-vs-Probe	Normal-vs-R2L	Normal-vs-U2R
Accuracy	0.90	0.92	0.79	0.98
Precision	0.99	0.87	0.99	0
Recall	0.77	0.69	0.07	0
F1-score	0.87	0.77	0.13	0
G-mean	0.87	0.81	0.26	0
MCC	0.80	0.72	0.231	-0.002
Balanced dataset	Normal-vs-DoS	Normal-vs-Probe	Normal-vs-R2L	Normal-vs-U2R
Accuracy	0.93	0.96	0.79	0.98
Precision	0.99	0.87	0.60	0.38
Recall	0.85	0.94	0.17	0.24
F1-score	0.91	0.90	0.26	0.30
G-mean	0.91	0.95	0.40	0.49
MCC	0.86	0.87	0.234	0.29

Although accuracy is the most common evaluation metric of machine learning models, it is an inappropriate metric for imbalanced classification because it does not distinguish between the number of correctly classified samples of the majority and minority classes. Therefore, it is obvious from Tables V, VI, and VII that there is a degradation in accuracy values of some attack categories after implementing CTGAN.

Moreover, the precision values of all classifiers decreased for some attacks categories, but improved for others. At the same time, recall values increased for all attack categories, except for DoS in the decision tree and KNN, where there was no improvement.

Precision and recall separately are not enough to evaluate the performance of the imbalanced classification. F1-score is

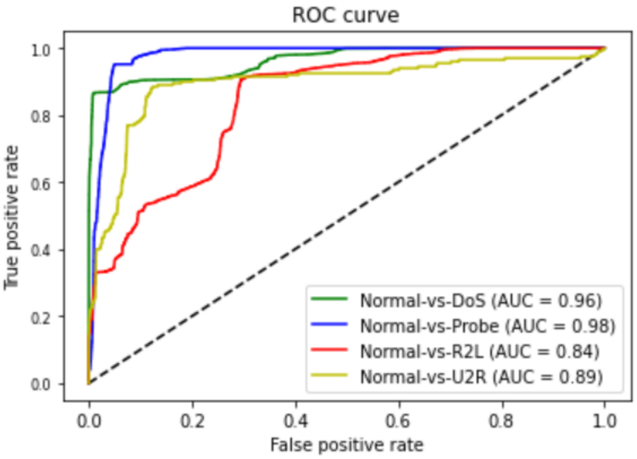

Fig. 4. ROC Curves of SVM on the Imbalanced Dataset.

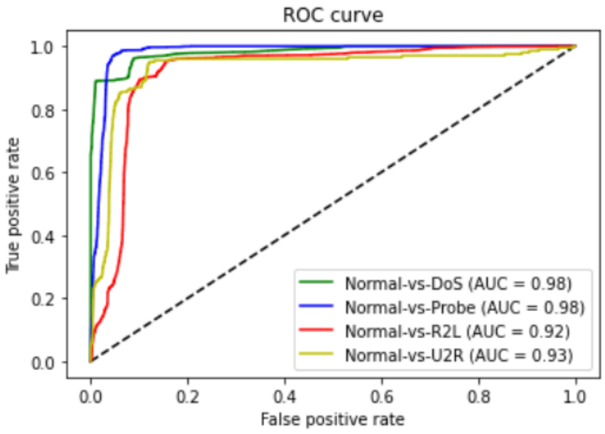

Fig. 5. ROC Curves of SVM on the Balanced Dataset.

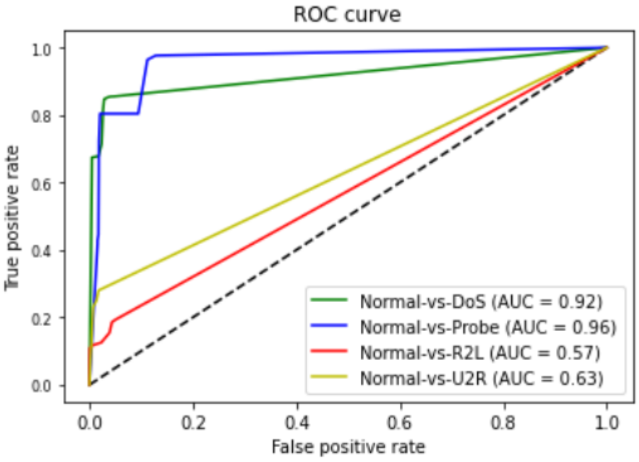

Fig. 6. ROC Curves of DT on the Imbalanced Dataset.

a balance of precision and recall; thus, it is a good metric for the imbalanced classification. Tables V, VI, and VII show an improvement in the F1-score values in all attack categories after using CTGAN, except for DoS with KNN, where there was no improvement.

TABLE VI. RESULTS OF ONE-VS-ONE DT CLASSIFICATION

Imbalanced dataset	Normal-vs-DoS	Normal-vs-Probe	Normal-vs-R2L	Normal-vs-U2R
Accuracy	0.91	0.94	0.80	0.98
Precision	0.96	0.91	0.98	0
Recall	0.82	0.80	0.11	0
F1-score	0.88	0.85	0.20	0
G-mean	0.89	0.88	0.33	0
MCC	0.81	0.821	0.291	0
Balanced dataset	Normal-vs-DoS	Normal-vs-Probe	Normal-vs-R2L	Normal-vs-U2R
Accuracy	0.91	0.96	0.80	0.94
Precision	0.98	0.88	0.68	0.22
Recall	0.82	0.91	0.21	0.73
F1-score	0.89	0.86	0.32	0.33
G-mean	0.90	0.92	0.45	0.83
MCC	0.83	0.828	0.299	0.37

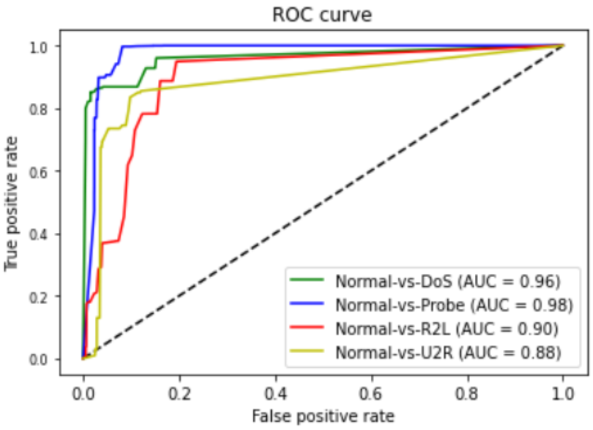

Fig. 7. ROC Curves of DT on the Balanced Dataset.

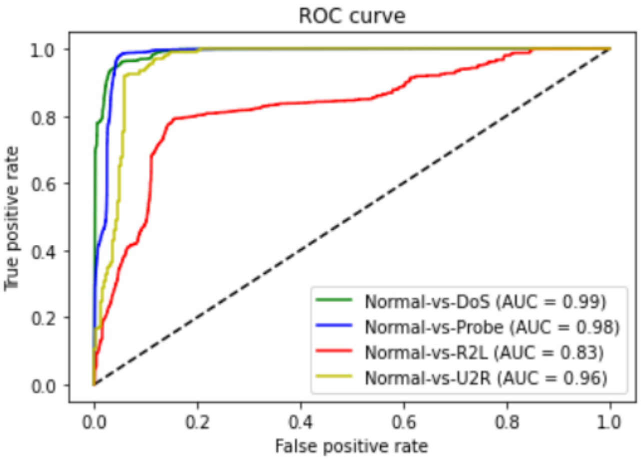

Fig. 8. ROC Curves of KNN on the Imbalanced Dataset.

G-mean is the balance between the accuracy of the algorithm on the majority class and the accuracy of the algorithm on the minority class. Hence, it is an appropriate evaluation metric for imbalanced classification. We notice from Tables V, VI, and VII that there is an improvement in G-mean values of all classifiers on all attack categories after using CTGAN, except for DoS with KNN there was no improvement.

TABLE VII. RESULTS OF ONE-VS-ONE KNN CLASSIFICATION

Imbalanced dataset	Normal-vs-DoS	Normal-vs-Probe	Normal-vs-R2L	Normal-vs-U2R
Accuracy	0.90	0.93	0.79	0.98
Precision	0.99	0.88	0.99	0.75
Recall	0.78	0.77	0.04	0.03
F1-score	0.87	0.82	0.07	0.06
G-mean	0.87	0.86	0.19	0.17
MCC	0.80	0.78	0.17	0.14
Balanced dataset	Normal-vs-DoS	Normal-vs-Probe	Normal-vs-R2L	Normal-vs-U2R
Accuracy	0.90	0.96	0.78	0.97
Precision	0.99	0.86	0.48	0.21
Recall	0.78	0.93	0.09	0.11
F1-score	0.87	0.89	0.15	0.15
G-mean	0.87	0.94	0.29	0.33
MCC	0.80	0.86	0.12	0.14

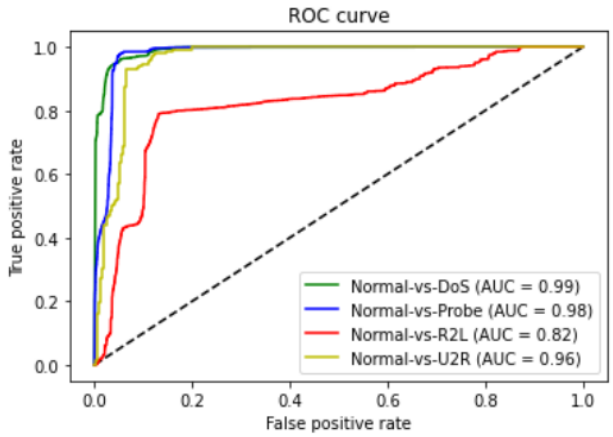

Fig. 9. ROC Curves of KNN on the Balanced Dataset.

While F1-score and G-mean are good evaluation metrics for imbalanced classification, MCC is more informative and reliable because it is determined based on the values of all of the cells of the confusion matrix. It is high only if all values of the confusion matrix are good. Moreover, MCC is the metric least affected by the imbalance issue. Thus, we notice from Tables V and VI that MCC values of SVM and decision tree increased on the balanced datasets for all attacks categories. On the other hand, there was no improvement in MCC values of KNN on DoS and U2R balanced datasets as shown in Table VII. Also, the MCC value of KNN was reduced on the balanced R2L dataset. One possible reason for this issue is that the KNN is less sensitive to the imbalance problem, which is consistent with [28]. Thus, oversampling the dataset did not improve the performance.

ROC curves and AUC scores are commonly used evaluation metrics for imbalanced classification. Fig. 4, Fig. 5 show the ROC curves and AUC scores of SVM on the imbalanced and balanced datasets of all attacks categories. We notice that AUC scores of all attack categories in the balanced datasets are clearly better than the AUC scores in the imbalanced datasets, which indicates an improved performance. Fig. 6 and Fig. 7 demonstrate the ROC curves and AUC scores of decision tree on the imbalanced and balanced datasets of all attacks categories. The AUC scores of all attacks categories are significantly higher after implementing CTGAN, especially for the rare classes (i.e., R2L and U2R). Fig. 8 and Fig. 9 depict the ROC curves and AUC scores of KNN on the imbalanced and balanced datasets of all attacks categories. We observed that the AUC scores of KNN did not show an improvement after the balancing process except with R2L, where there was a slight reduction.

Based on the F1-score, G-mean, MCC and AUC values obtained, using CTGAN to generate synthetic samples has improved the performance of the SVM and decision tree models. At the same time, we notice that the positive impact of CTGAN was not obvious on KNN, instead it causes a slight reduction in some evaluation metrics. This is due to the insensitivity of the KNN to the imbalance problem.

VI. CONCLUSION

Most of the intrusion detection datasets are imbalanced due to the natural difference between the number of intrusive and normal samples. There are many techniques to deal with the imbalance problem. One of these techniques is oversampling (e.g., ROS, SMOTE, and GAN-based methods). In this study, the focus is on CTGAN which can generate more samples of a specific class. CTGAN was applied on NSL-KDD to increase the number of intrusive samples and make the dataset balanced. The effectiveness of CTGAN was evaluated by running SVM, decision tree, and KNN models on the dataset before and after using CTGAN. Experiments using various types of attacks and one-vs-one classification were conducted in this study. CTGAN proved its effectiveness in generating synthetic data resembling real intrusions to improve the performance of SVMs and decision trees on the imbalanced datasets in terms of F1-score, G-mean, MCC, and AUC values. On the other hand, CTGAN did not show an improvement in KNN performance because it is less sensitive to class imbalance problem. For future work, more machine learning and deep learning models could be used. Furthermore, other imbalance techniques could be evaluated and compared with CTGAN.

ACKNOWLEDGMENT

The authors would like to thank King Fahd University and the University of Hafr Al Batin for support during this work.

REFERENCES

[1] R. Vinayakumar, M. Alazab, K. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, “Deep learning approach for intelligent intrusion detection system,” *IEEE Access*, vol. 7, pp. 41 525–41 550, 2019.

[2] H. Kaur, H. S. Pannu, and A. K. Malhi, “A systematic review on imbalanced data challenges in machine learning: Applications and solutions,” *ACM Computing Surveys (CSUR)*, vol. 52, no. 4, pp. 1–36, 2019.

[3] L. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni, “Modeling tabular data using conditional gan,” in *Advances in Neural Information Processing Systems*, 2019, pp. 7335–7345.

[4] A. Shafee, M. Baza, D. A. Talbert, M. M. Fouda, M. Nabil, and M. Mahmoud, “Mimic learning to generate a shareable network intrusion detection model,” in *2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 2020, pp. 1–6.

[5] J. Lee and K. Park, “Gan-based imbalanced data intrusion detection system,” *Personal and Ubiquitous Computing*, pp. 1–8, 2019.

[6] C. Yin, Y. Zhu, J. Fei, and X. He, “A deep learning approach for intrusion detection using recurrent neural networks,” *IEEE Access*, vol. 5, pp. 21 954–21 961, 2017.

[7] H. Zhang, L. Huang, C. Q. Wu, and Z. Li, “An effective convolutional neural network based on smote and gaussian mixture model for intrusion detection in imbalanced dataset,” *Computer Networks*, p. 107315, 2020.

[8] S. E. Gómez, L. Hernández-Callejo, B. C. Martínez, and A. J. Sánchez-Esguevillas, “Exploratory study on class imbalance and solutions for network traffic classification,” *Neurocomputing*, vol. 343, pp. 100–119, 2019.

[9] Z. Hosenie, R. Lyon, B. Stappers, A. Mootoovaloo, and V. McBride, “Imbalance learning for variable star classification,” *Monthly Notices of the Royal Astronomical Society*, vol. 493, no. 4, pp. 6050–6059, 2020.

[10] Y. Sun, M. Li, L. Li, H. Shao, and Y. Sun, “Cost-sensitive classification for evolving data streams with concept drift and class imbalance,” *Computational Intelligence and Neuroscience*, vol. 2021, 2021.

[11] X. Zhang, J. Ran, and J. Mi, “An intrusion detection system based on convolutional neural network for imbalanced network traffic,” in *2019 IEEE 7th International Conference on Computer Science and Network Technology (ICCSNT)*. IEEE, 2019, pp. 456–460.

[12] A. Aggarwal, M. Mittal, and G. Battineni, “Generative adversarial network: An overview of theory and applications,” *International Journal of Information Management Data Insights*, vol. 1, no. 1, p. 100004, 2021.

[13] S. Park and H. Park, “Performance comparison of multi-class svm with oversampling methods for imbalanced data classification,” in *International Conference on Broadband and Wireless Computing, Communication and Applications*. Springer, 2020, pp. 108–119.

[14] M. Mirza and S. Osindero, “Conditional generative adversarial nets,” *arXiv preprint arXiv:1411.1784*, 2014.

[15] R. Abdulhammed, M. Faezipour, A. Abuzneid, and A. AbuMallouh, “Deep and machine learning approaches for anomaly-based intrusion detection of imbalanced network traffic,” *IEEE sensors letters*, vol. 3, no. 1, pp. 1–4, 2018.

[16] A. A. ALFRHAN, R. H. ALHUSAIN, and R. U. Khan, “Smote: Class imbalance problem in intrusion detection system,” in *2020 International Conference on Computing and Information Technology (ICCIT-1441)*. IEEE, 2020, pp. 1–5.

[17] I. Yilmaz, R. Masum, and A. 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)*. IEEE, 2020, pp. 25–30.

[18] S. Huang and K. Lei, “Igan-ids: An imbalanced generative adversarial network towards intrusion detection system in ad-hoc networks,” *Ad Hoc Networks*, vol. 105, p. 102177, 2020.

[19] P. Bedi, N. Gupta, and V. Jindal, “Siam-ids: Handling class imbalance problem in intrusion detection systems using siamese neural network,” *Procedia Computer Science*, vol. 171, pp. 780–789, 2020.

[20] B. Setiawan, S. Djanali, and T. Ahmad, “Analyzing the performance of intrusion detection model using weighted one-against-one support vector machine and feature selection for imbalanced classes,” *Int. J. Intell. Eng. Syst*, vol. 13, pp. 151–160, 2020.

[21] S. M. Gaffer, M. E. Yahia, and K. Ragab, “Genetic fuzzy system for intrusion detection: Analysis of improving of multiclass classification accuracy using kddcup-99 imbalance dataset,” in *2012 12th International Conference on Hybrid Intelligent Systems (HIS)*. IEEE, 2012, pp. 318–323.

[22] C. Elkan, “Results of the kdd’99 classifier learning,” *ACM SIGKDD Explorations Newsletter*, vol. 1, no. 2, pp. 63–64, 2000.

[23] M.-Y. Su, C.-Y. Lin, S.-W. Chien, and H.-C. Hsu, “Genetic-fuzzy association rules for network intrusion detection systems,” in *Proc. IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2011)*, 2011, pp. 2046–2052.

[24] C.-H. Tsang, S. Kwong, and H. Wang, “Genetic-fuzzy rule mining approach and evaluation of feature selection techniques for anomaly intrusion detection,” *Pattern Recognition*, vol. 40, no. 9, pp. 2373–2391, 2007.

[25] J. Gomez and D. Dasgupta, “Evolving fuzzy classifiers for intrusion detection,” in *Proc. IEEE Workshop on Information Assurance*, vol. 6, no. 3, 2002, pp. 321–323.

[26] R. Agarwal and M. V. Joshi, “Pnrule: A new framework for learning classifier models in data mining (a case-study in network intrusion detection),” in *Proc SIAM International Conference on Data Mining*. SIAM, 2001, pp. 1–17.

[27] A. Alabdallah and M. Awad, “Using weighted support vector machine to address the imbalanced classes problem of intrusion detection system,” *KSII Transactions on Internet and Information Systems (TIIS)*, vol. 12, no. 10, pp. 5143–5158, 2018.

[28] P. Thanh Noi and M. Kappas, “Comparison of random forest, k-nearest neighbor, and support vector machine classifiers for land cover classification using sentinel-2 imagery,” *Sensors*, vol. 18, no. 1, p. 18, 2017.