A Comprehensive Multi-Stage NIDS Utilizing GAN For Data Balancing

Omar Awadallah, Sarah Azzabi, Yazan Aref
Electrical and Computer Engineering Department
Western University
London, Canada
{oawadall, sazzabi, yaref}@uwo.ca

***Abstract*—The rapid proliferation of internet-connected devices with varying capabilities and security measures introduces various security vulnerabilities, which induce attackers to exploit them to gain access to the network's resources, leaving room for newer and more sophisticated attacks to emerge. Thus, it is mandatory to promptly design and develop fast and reliable *Network Intrusion Detection Systems* (NIDSs) to mitigate such scenarios.**

In this work, we propose developing an efficient NIDS by utilizing *Generative Adversarial Networks* (GAN) to overcome the issue of class imbalance and a multi-stage classification architecture that includes a binary classifier followed by a multi-class classifier to ensure faster and more efficient performance. Our proposed approach is evaluated on the CIC-IDS2018 dataset containing various up-to-date attacks. Our evaluation results verify the efficient performance of the proposed GAN implementation and the hierarchical multi-stage architecture. The paper concludes that the proposed approach resulted in a value of around 99% for the f1-score, precision, and recall metrics.

***Keywords*— Network Intrusion Detection, Generative adversarial networks, Ensemble Learning, Anomaly Detection, Imbalanced Data**

I. Introduction

The advancement of technology and the internet has transformed the way humans communicate, work, and share information in a fast and efficient manner. In the current time, the use of the internet became a necessity among individuals and businesses to transfer and store sensitive information. However, with the advancement of the internet, the risk of security threats to access sensitive information for nefarious purposes has increased. Such threats pose a serious risk to the economy, user privacy, or national security [1]. The current threats have served as a motivation for this work, and other existing works (e.g. [2], [3]), to develop a Network Intrusion Detection System (NIDS) that can effectively mitigate them.

In this paper, data about network intrusions is used to predict whether or not network traffics are benign or attacks, such predictions can help control the network and ensure the privacy and safety of sensitive data. Sampling techniques are used to mitigate the effect of imbalanced data, and feature selection algorithms are used to remove any irrelevancy or redundancy. After feature selection, *General Adversarial Networks* (GAN) is used to generate new data to help the machine learning model learn the distribution quickly. A multi-stage model is used to classify the network traffics, the model stage consists of an ensemble model followed by a random forest model to classify normal traffics and filter them out then classify the type of attacks, this ensures a faster prediction process with higher accuracy, a faster prediction is important because it is crucial to detect nefarious attacks as fast as possible to stop them. In what follows we start with discussing related work, the selected dataset description, the proposed methodology, and the achieved results, and then we discuss our next steps.

II. Related Work

A handful of the previous literature has focused on developing NIDS through various techniques and datasets. *Javaid et. al.,* [4] utilized *Softmax-Regression* deep learning method to classify network packets. They used the KDDCUP99 dataset, which is a widely recognized benchmark for NIDS. The authors established standard evaluation metrics for accuracy, including precision, recall, and f-measure. Their NIDS outperformed previously implemented NIDSs for normal/anomaly detection when evaluated on the test data. However, given the constantly evolving nature of technology and the emergence of new cyberattacks, benchmarks must also evolve. Moreover, since the KDDCUP99 is outdated, it is imperative to adopt newer and more comprehensive datasets.

Bhati et. al., [5] conducted an analytical study on *Support Vector Machine* (SVM)-based NIDS. They also employed the KDDCUP99 and evaluated the performance based on the accuracy, confusion matrix, as well as true-positive and false-positive rates. The authors employed four different SVM techniques, including linear, quadratic, fine Gaussian, and medium Gaussian SVMs. They found that the fine Gaussian SVM provided the best accuracy and least error for NIDS.

As NID datasets, by and large, suffer from class imbalance, researchers have been committed to addressing this problem. In particular, *Liu el. al.,* [6] introduced the under-sampling technique method as a conventional and frequently used technique for treating class imbalance. This method entails excluding

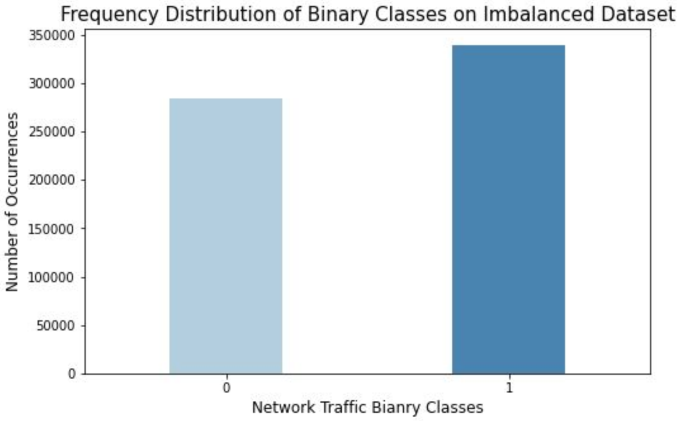


Fig. 1. Distribution of benign and combined attack classes

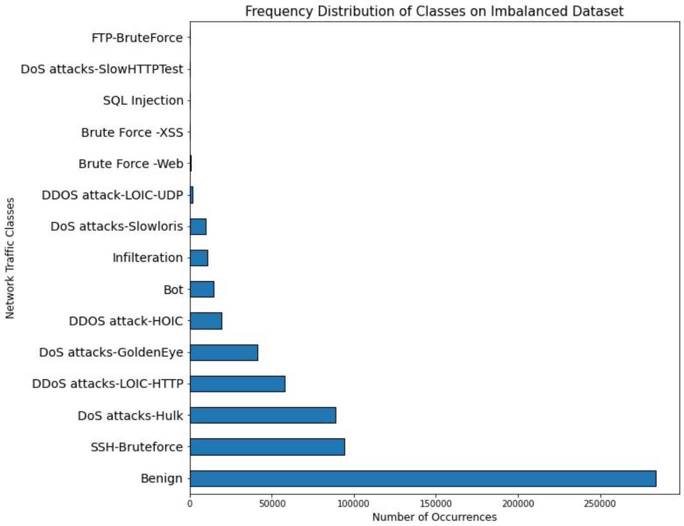


Fig. 2. Distribution of benign and individual attack classes

a chosen subset of the majority class during model training to achieve a balanced dataset. Nonetheless, the approach's drawback is that it raises the possibility of losing significant unique instances that are vital for precise detection.

In [7], *Lee et. al.,* investigated the application of the Generative Adversarial Network (GAN) model to tackle the issue of data imbalance in intrusion data. GAN is a recent deep learning method that generates new artificial data by utilizing existing data to ensure they are similar. The authors demonstrated that using Random Forest to classify intrusions after GAN data balancing led to better performance compared to classifying without GAN balancing. They employed the CIC-IDS2017, which is an earlier version of the dataset used in our study.

Our work will be benchmarked against [8]. Where *Soltani et. al.,* used the same dataset and metrics to evaluate the performance of their implementation. They employed a deep learning-based approach where they used *Long Sort-Term Memory* (LSTM) to classify attacks. Noting that, in their work, attacks that fall under the same category (such as DoS HOIC and DoS HTTP) were combined as a single category (DoS).

III. Dataset

In this work, the CSE-CIC-IDS2018 [9] dataset, provided by the University of New Brunswick, is utilized to train and test the proposed design. The dataset contains descriptions of network and protocol intrusions to facilitate an in-depth analysis of network attacks.

The data was collected by executing various types of attacks (e.g. BruteForce, DDoS, SQL injection, Infiltration, etc.) on different machine IPs, these attacks were performed for ten non-consecutive days and the raw data, such as network traffic (Pcap files) and event logs, were captured and organized per day. The raw data was processed using CICFlowMeter-V3, a network traffic flow generator and analyzer, and 78 features (e.g. flow duration, number of packets with ACK, total packets in the forward direction, etc.) were extracted.

The original dataset is divided into ten parts, each representing a different day and containing data on various attack classes. The different parts were merged together into one dataset before starting the preprocessing step, the final merged dataset has a total of 623,356 samples and 79 columns, 78 columns represent the features and one column represents the label that classifies traffic into benign or attacks, it also specifies the type of the attack for a total of 14 different attacks.

A thorough analysis of the dataset is conducted to understand the nature of the data and decide on the necessary preprocessing techniques that need to be implemented to train an accurate unbiased NIDS system. The analysis demonstrated the presence of severe data imbalance. Figure 1 shows the distribution between the benign traffic and attacks, it is clear that the attack class has more samples than the benign class, this imbalance distribution is not severe because it weighs the attacks more which is important since one attack is more severe than one benign traffic. However, the imbalance among the types of attack differs significantly as seen in Fig. 2, this will produce an inaccurate model that will be biased towards a specific type of attack, the SHH-Bruteforce attack, which negatively affects the overall performance of the system. Therefore, data balancing using GANs and sampling techniques are needed to achieve an unbiased model.

IV. Methodology

Our proposed methodology is demonstrated in Fig. 3, which composes of three primary phases. The first step involves preprocessing and selecting the most important features from the CIC-IDS18 dataset. The second phase is concerned with treating the issue of class imbalance by applying a GAN model to create synthetic examples. Finally, the classification phase consists of a binary classifier, followed by a multi-stage classifier.

A. Preprocessing

For the preprocessing phase, we first cleaned the dataset by removing instances that include null or infinity values. Next,

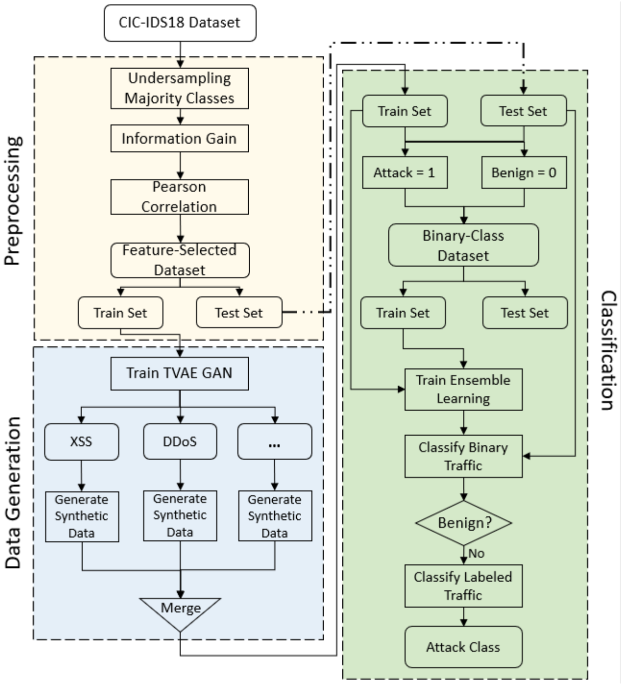


Fig. 3. Proposed Model

we processed with feature selection, where we filtered through the dataset's features and kept the most important ones. Our feature selection method was based on calculating the *Information Gain* (IG) to eliminate redundant features, followed by *Pearson Correlation* (PC) to only preserve highly correlated features. IG simply measures how much a specific feature contributes to classifying the target outcome by calculating the entropy reduction when splitting the data based on the specific feature. PC is a -1 to 1 value that signifies how two variables are linearly related.

B. Data Generation

Class imbalance in NIDS datasets is a widely known problem that poses a considerable quandary to ML and DL models, this is due to the fact that most ML and DL models tend to get biased towards classes with the highest distribution in the dataset, which decreases models' performance [10]. To overcome this significant dilemma, research has been recently directed to employ GAN models. GANs aim to be trained on a dataset to generate synthetic data that correspond to real features of the training data.

In 2014, *Goodfellow et. al* [11] introduced GAN as a neural network architecture that is able to generate new data while maintaining the characteristics of the actual data, by applying a two-player game that contains the *Generator* ($\mathcal{G}$) and the *Discriminator* ($\mathcal{D}$). The process starts by learning the distribution $\mathcal{P}_g$ of the actual data $\mathcal{X}$, and loading random noise to $\mathcal{G}$, it generates new fake samples $\mathcal{G}(\mathcal{Z})$ that have a very similar distribution to the actual data space $\mathcal{X}$. The role of $\mathcal{D}$ is to predict whether the new instances are real or fake by calculating the probability when inputting a real sample $\mathcal{D}(\mathcal{X})$ or a generated sample $\mathcal{D}(\mathcal{G}(\mathcal{Z}))$ and adjust $\mathcal{G}$'s wights based on the result. This competition derives the model as a whole to enhance its performance when generating synthetic data.

After conducting a comparison of the most utilized GAN models (section. V), we employed the *Tabular Variational Autoencoder* (TVAE) [12] to generate new instances for each of the minority classes. We used a *Multi-GAN* model, which indicates that we applied TVAE to each minority class separately in the training set, leaving the test set untouched. This preserves the actual characteristics of each class.

C. Classification

In this subsection, we describe the last phase of our implementation, where the classification process occurs. We first create a copy of the dataset and adjust the labels to either "Attack" or "Benign". Using the *Ensemble Learning* (EL) model, we classify instances in the test set to one of the two classes. Afterward, instances classified as "Attack" are passed to the *Random Forest* (RF) model to predict the class/type of attack out of the 14 classes. Noting that, RF is also trained to classify "Benign" traffic in case of any incorrect predictions from the EL model.

As EL is concerned with classifying data into only two classes, by employing EL first to classify whether an instance is an attack or not, and only passing attack instances to the RF model, we potentially reduce the overall complexity of the whole system, which results in a faster, and better performance.

1) Ensemble Learning:

Our EL model consisted of only two ML algorithms, namely a *Decision Tree* (DT) model, and a *Logistic Regression* (LR) model, as both models were proven to be very efficient and fast when employed to binary classification problems [13].

2) Random Forest:

Since there have been a lot of successes from previous researchers as in [14] in indicating RF's excellent achievable performance, we employed the RF model in our architecture to classify instances labeled as "Attack" from the EL model. We also trained our RF model on the full dataset, in order to mitigate any incorrect predictions from the EL model (inaccurately classifying "Benign" instances as "Attack").

There are various metrics that can be used to evaluate the performance of a trained model, and for our work, we have selected precision (PR), recall (RC), and F1 score (F1). These metrics can be calculated based on the values in a confusion matrix, where TP refers to true positives, FP refers to false positives, TN refers to true negatives, and FN refers to false negatives. The equations for these metrics are as follows:

$$PR = \frac{TP}{TP + FP} \tag{1}$$

$$RC = \frac{TP}{TP + FN} \tag{2}$$

$$F1 - Score = \frac{2 \times PR \times RC}{PR + RC} \tag{3}$$

Precision (PR) represents the ratio of correctly classified instances to all classifications. It is a useful tool for evaluating the effectiveness of an IDS in generating reliable security classifications. Recall (RC) is an important metric for NIDS as it measures the ratio of detected attacks to the actual number of attacks. This metric is particularly valuable because it provides insight into the ability of the system to correctly identify attacks, which is a critical aspect of IDS performance. Finally, The F1-Score is a metric that seeks to balance the significance of precision and recall by computing their harmonic mean.

V. Results & Analysis

Before conducting the model training, an initial evaluation was carried out, which involved removing irrelevant and redundant data through feature selection. This resulted in a total of 23 relevant features being retained for the subsequent analysis. For the data generation stage, three types of GAN architecture (CT-GAN, TVAE-GAN, Copula-GAN) provided by the Synthetic Data Vault (SDV) Python library are considered and compared on the training set to choose the best GAN network. The similarity score is used as the evaluation metric to compare the three GANs, the similarity score evaluates how similar is the generated synthetic data to the real data. From table I, it is observed that the TVAE-GAN model has the highest similarity score of 0.80, hence it is chosen to perform the data sampling.

TABLE I
Sampling performance between the three GANs

Model	***Similarity Score***
CT-GAN model	0.747
Copula-GAN model	0.791
TVAE-GAN model	0.800

TABLE II
Sampling performance between Multi-TVAE-GAN and Single-TVAE-GAN

Model	***Similarity Score***
Multi-TVAE-GAN models	0.9020
Single-TVAE-GAN model	0.8000

To further improve the performance of the TVAE model and ensure that the produced attack samples conform to the boundary of the original attack samples, we trained a total of 10 TVAE models for each selected attack category. To accomplish this, we separated each attack category from the original training set and created a distinct subset for training the TVAE-GAN model. For example, the Infiltration attack class has its own homogeneous dataset comprising instances belonging exclusively to this particular category. The reason for separating the attacks from the original training dataset and training a separate model for each category during the TVAE-GAN training was to guarantee that the generator could generate the features of a particular category accurately, rather than generating features that could be a blend of all 10 categories.

The assessment showed that the multiple GAN models outperform the single GAN as depicted in table II, where utilizing a separate GAN for each attack class resulted in a similarity score of roughly 0.9, while a single GAN trained on all attack classes achieved a score of about 0.8. Following the GAN models training, A total of 155,000 attack samples are generated using the trained models and merged to the original training set, Fig. 4 shows the training set class distribution after implementing GAN to the data where the distribution between attacks and benign traffic is similar, and the samples among the attack types are fairly distributed, this result demonstrates that the data is ready for the training process.

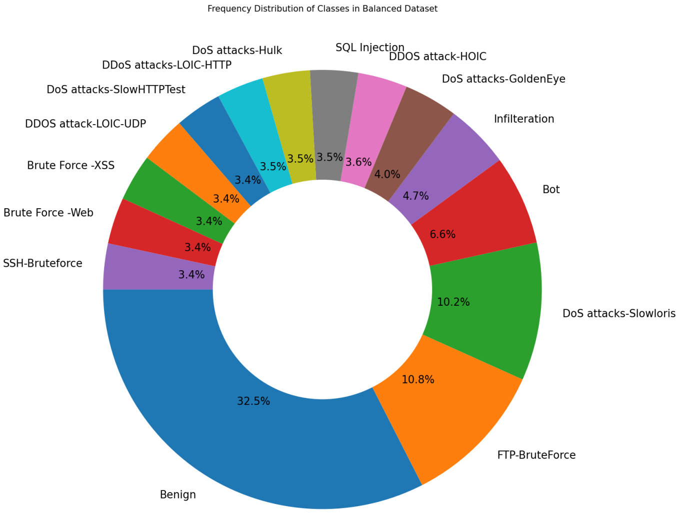


Fig. 4. Distribution of classes in balanced dataset

Once the data was processed to train the NIDS model, the performance of the ensemble model for binary classification was evaluated individually on the binary test set, then the multi-stage model (ensemble + RF) was evaluated on the multi-class test set. The ensemble model performed well on the binary test set as it classified a large number of true positives and negatives correctly with around 96% score for f1-score, precision, and recall metrics as seen in III and Fig.5 respectively. The multi-stage model performed well overall and achieved around 99% for f1-score, precision, and recall as seen in table IV.

TABLE III
Evaluation for Ensemble Model on Binary Set

Metric	***Score***
F1-Score	0.96134
Precision	0.96150
Recall	0.96138

TABLE IV
Evaluation for Multi-Stage Model on Multi-class Set

Metric	***Score***
F1-Score	0.98991
Precision	0.99118
Recall	0.98897

However, when assessing the performance of the model on individual classes, the model had poor performance in

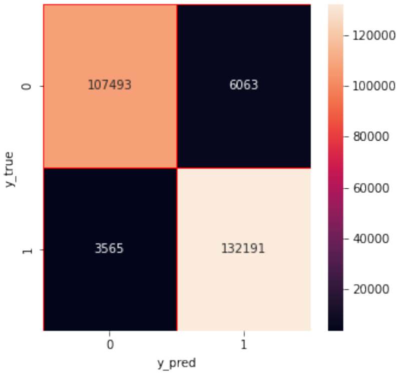


Fig. 5. Ensemble Model Confusion Matrix

predicting *SlowHTTPTest DoS attacks*, *FTP BruteForce*, and *Infiltrations* as seen in table V.

TABLE V
PRECISION SCORES FOR INDIVIDUAL CLASSES

Class	***Precision***
Benign	0.91
Bot	1.00
BruteForce-Web	0.93
BruteForce-XSS	1.00
DDoS attack-HOIC	1.00
DDoS attack-LOIC-UDP	0.97
DDoS attack-LOIC-HTTP	1.00
DoS attack-GoldenEye	1.00
DoS attack-Hulk	1.00
DoS attack-SlowHTTPTest	0.19
DoS attack-Slowloris	1.00
FTP-BruteForce	0.18
Infiltration	0.28
SQL Injection	0.69
SSH-BruteForce	1.00

The confusion matrix was investigated to assess this behavior, and it was found that the model has misclassified large samples of infiltrations as benign traffic and vice-versa as seen in Fig. 6 marked in blue. This can be due to the nature of the infiltration attacks, this type of attack is not an attack on the server itself but rather it is a way of tricking victims into voluntarily providing their credentials to attackers using malicious links, therefore its network features are similar to benign traffic. Also, the model clearly is unable to distinguish between *SlowHTTPTest DoS attacks* and *FTP BruteForce* as it misclassified the same number of samples for both attacks (marked in yellow). When the dataset set was examined for both attacks (labels 9 ad 11) in Fig. 7, it was found that only two features are distinct and there is a lack of variation in the features between the two classes with 95.1% similarity which made them hard to be distinguishable by the model.

VI. CONCLUSION & NEXT STEPS

With the significant advancements in networking technology and the exponential increase in data volume, attackers are continuously devising new ways to exploit vulnerabilities in networking systems. This research introduced a new approach to Network Intrusion Detection Systems that tackles the drawbacks of previous NIDSs by implementing innovative solutions that significantly enhance their performance. Our work has utilized the recent CIC-IDS2018 dataset to cover as many recent attacks as possible. The data was cleaned from any null or infinite values, irrelevant features were filtered out using *Information Gain*, and redundant features were dropped using *Pearson Correlation*. A multi-TVAE-GAN model was used for sampling to balance the classes. A multi-stage NIDS was developed to detect nefarious network attacks, this system resulted in 99% precision for the overall classification of the benign and 14 attack classes. However, the system had a low precision value when classifying *SlowHTTPTest DoS attack* and *FTP BruteForce* as it could not distinguish between them because of low variance in their features, it has also misclassified some *infiltrations* as benign traffic.

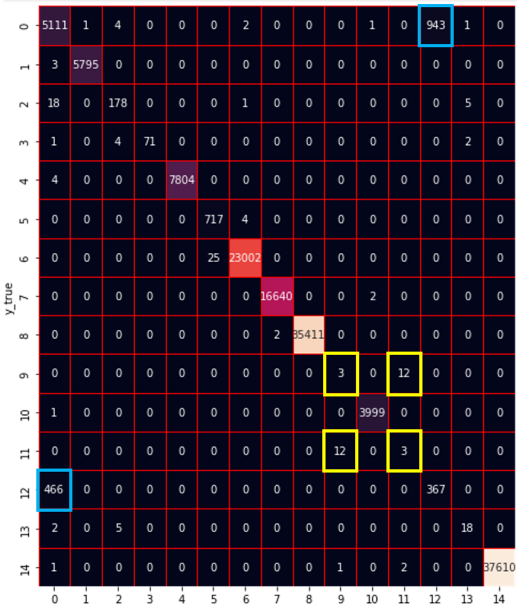
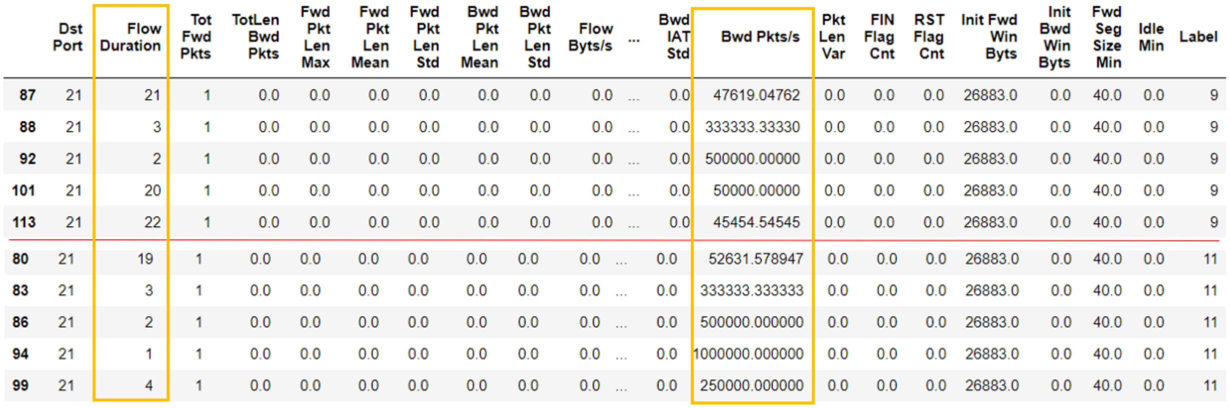


Fig. 6. Multi-Stage Model Confusion Matrix

The following steps will concentrate on enhancing attack classification through various techniques, including feature engineering. This strategy intends to generate additional distinct features for *SlowHTTPTest DoS attacks* and *FTP BruteForce*. Model tuning will also be considered to identify the optimal parameters to improve accuracy. Finally, the ensemble structure will be modified by experimenting with other models that were not used initially.

VII. CONTRIBUTION STATEMENT

To indicate each member's roles in the project, for the preprocessing part, Yazan and Omar were responsible for preprocessing and feature selection. Omar and Sarah collaborated on the utilization and assessment of the GAN model. Finally, Yazan and Sarah worked on the classification model, and the three group members equally collaborated on writing the final report.



	Dst Port	Flow Duration	Tot Fwd Pkts	TotLen Bwd Pkts	Fwd Pkt Len Max	Fwd Pkt Len Mean	Fwd Pkt Len Std	Bwd Pkt Len Mean	Bwd Pkt Len Std	Flow Byts/s	...	Bwd IAT Std	Bwd Pkts/s	Pkt Len Var	FIN Flag Cnt	RST Flag Cnt	Init Fwd Win Byts	Init Bwd Win Byts	Fwd Seg Size Min	Idle Min	Label
87	21	21	1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	47619.04762	0.0	0.0	0.0	26883.0	0.0	40.0	0.0	9
88	21	3	1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	333333.33330	0.0	0.0	0.0	26883.0	0.0	40.0	0.0	9
92	21	2	1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	500000.00000	0.0	0.0	0.0	26883.0	0.0	40.0	0.0	9
101	21	20	1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	50000.00000	0.0	0.0	0.0	26883.0	0.0	40.0	0.0	9
113	21	22	1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	45454.54545	0.0	0.0	0.0	26883.0	0.0	40.0	0.0	9
80	21	19	1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	52631.578947	0.0	0.0	0.0	26883.0	0.0	40.0	0.0	11
83	21	3	1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	333333.333333	0.0	0.0	0.0	26883.0	0.0	40.0	0.0	11
86	21	2	1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	500000.000000	0.0	0.0	0.0	26883.0	0.0	40.0	0.0	11
94	21	1	1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	1000000.000000	0.0	0.0	0.0	26883.0	0.0	40.0	0.0	11
99	21	4	1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	250000.000000	0.0	0.0	0.0	26883.0	0.0	40.0	0.0	11

Fig. 7. Features for labels 9 and 11

VIII. References

[1] M. Ahmed, A. Mahmood, and J. Hu, *A survey of network anomaly detection techniques*, Dec. 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1084804515002891.

[2] W. Liang, S. Xie, D. Zhang, X. Li, and K.-c. Li, *A mutual security authentication method for rfid-puf circuit based on deep learning*, 2021. DOI: 10.1145/3426968.

[3] Q. Tian, D. Han, K.-C. Li, X. Liu, L. Duan, and A. Castiglione, *An intrusion detection approach based on improved deep belief network and lightgbm*, May 2022. DOI: 10.1109/iscsic57216.2022.00020.

[4] A. Javaid, Q. Niyaz, W. Sun, and M. Alam, "A deep learning approach for network intrusion detection system," in *Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS)*, 2016, pp. 21–26.

[5] B. S. Bhati and C. S. Rai, "Analysis of Support Vector Machine-based Intrusion Detection Techniques," en, *Arabian Journal for Science and Engineering*, vol. 45, no. 4, pp. 2371–2383, Apr. 2020, ISSN: 2193-567X, 2191-4281. DOI: 10.1007/s13369-019-03970-z. [Online]. Available: http://link.springer.com/10.1007/s13369-019-03970-z (visited on 12/15/2022).

[6] Xu-Ying Liu, Jianxin Wu, and Zhi-Hua Zhou, "Exploratory Undersampling for Class-Imbalance Learning," en, *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 39, no. 2, pp. 539–550, Apr. 2009, ISSN: 1083-4419. DOI: 10.1109/TSMCB.2008.2007853. [Online]. Available: http://ieeexplore.ieee.org/document/4717268/ (visited on 12/20/2022).

[7] J. Lee and K. Park, "GAN-based imbalanced data intrusion detection system," en, *Personal and Ubiquitous Computing*, vol. 25, no. 1, pp. 121–128, Feb. 2021, ISSN: 1617-4909, 1617-4917. DOI: 10.1007/s00779-019-01332-y. [Online]. Available: http://link.springer.com/10.1007/s00779-019-01332-y (visited on 10/19/2022).

[8] M. Soltani, M. J. Siavoshani, and A. H. Jahangir, *A content-based deep intrusion detection system - international journal of information security*, Sep. 2021.

[9] *Cse-cic-ids-2018*. [Online]. Available: https://www.unb.ca/cic/datasets/ids-2018.html.

[10] L. Vu, C. T. Bui, and Q. U. Nguyen, "A Deep Learning Based Method for Handling Imbalanced Problem in Network Traffic Classification," en, in *Proceedings of the Eighth International Symposium on Information and Communication Technology*, Nha Trang City Viet Nam: ACM, Dec. 2017, pp. 333–339, ISBN: 978-1-4503-5328-1. DOI: 10.1145/3155133.3155175. [Online]. Available: https://dl.acm.org/doi/10.1145/3155133.3155175 (visited on 10/16/2022).

[11] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, *et al.*, *Generative Adversarial Networks*, en, arXiv:1406.2661 [cs, stat], Jun. 2014. [Online]. Available: http://arxiv.org/abs/1406.2661 (visited on 10/16/2022).

[12] L. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni, *Modeling Tabular data using Conditional GAN*, en, arXiv:1907.00503 [cs, stat], Oct. 2019. [Online]. Available: http://arxiv.org/abs/1907.00503 (visited on 12/18/2022).

[13] C. A. Ul Hassan, M. S. Khan, and M. A. Shah, "Comparison of machine learning algorithms in data classification," *2018 24th International Conference on Automation and Computing (ICAC)*, 2018. DOI: 10.23919/iconac.2018.8748995.

[14] L. Grinsztajn, E. Oyallon, and G. Varoquaux, *Why do tree-based models still outperform deep learning on tabular data?* 2022. DOI: 10.48550/ARXIV.2207.08815. [Online]. Available: https://arxiv.org/abs/2207.08815.