4. Interdisciplinary Conference on Electrics and Computer (INTCEC 2024)
11-13 June 2024, Chicago-USA

Enhancing Intrusion Detection through Deep Learning and Generative Adversarial Network

Md Habibur Rahman*, Leo Martinez III*, Avdesh Mishra†, Mais Nijim, Ayush Goyal and David Hicks
Department of Electrical Engineering and Computer Science
Texas A&M University-Kingsville, Kingsville, TX 78363, USA
Email: {md_habibur.rahman, leo.martinez}@students.tamuk.edu, {avdesh.mishra, mais.nijim, ayush.goyal, david.hicks}@tamuk.edu
*Equal Authorship. †Author to whom correspondence should be addressed.

***Abstract*—Network behavior during intrusion deviates from the standard, which can be identified by establishing a baseline of typical activity. Accurate detection of diverse attack classes with machine learning relies on having adequate representative samples for each class. Typically, highly imbalanced datasets like the NSL-KDD led to a biased model favoring the dominant classes. To address the challenge, we employ a Conditional Tabular Generative Adversarial Network (CTGAN) for generating synthetic samples to balance the dataset effectively. Later on, a deep neural network with a final layer comprising 4 neurons is applied for multi-class classification. The proposed method is compared with state-of-the-art approaches that utilize Conditional Generative Adversarial Network (CGAN) and Wasserstein Conditional Generative Adversarial Network (WCGAN) sampling techniques is found to yield an average improvement of 105.93%, 56.37%, and 80.81% based on Precision, Recall, and F1 Score.**

***Index Terms*—Intrusion detection, Minority oversampling, Deep neural network, Cybersecurity/Cyber-attack, Conditional Tabular Generative Adversarial Network, NSL-KDD**

I. Introduction

In the rapidly evolving digital landscape, safeguarding network infrastructure is a paramount concern for global organizations. The escalating frequency and sophistication of cyber threats necessitate innovative approaches to fortify cyber defenses. Machine learning (ML) coupled with network traffic classification emerges as a proactive solution to combat these challenges [1], [2]. A key obstacle in intrusion detection is the timely and accurate detection of network-based anomalies, which can indicate potential security breaches. Various technologies address the challenge of network-based anomaly intrusion detection. However, conventional methods such as signature-based detection may falter against modern sophisticated attacks [3]. Maintaining an updated signature database further complicates defense against rapidly evolving threats.

Intrusion detection systems (IDS) represent a specialized category designed to monitor and analyze network traffic for signs of malicious activity [4], [5]. These systems employ different statistical approaches to detect intrusions, with three prominent methods being signature-based detection, anomaly-based detection, and behavior-based detection [6]. Signature-based detection relies on a predefined set of patterns associated with known threats. It can efficiently identify known threats but can be bypassed by polymorphic attacks. While anomaly-based detection seeks to identify deviations from normal network behavior. By establishing a baseline of regular activity, any abnormal activities are flagged as potential intrusions. However, this approach may produce false positives or negatives, and most problematic, defining a precise "normal" can be challenging in dynamic network environments. Moreover, behavior-based detection focuses on the behavior of network entities and users. It observes patterns of activity and identifies abnormal behavior based on deviations from expected network behavior. While providing a more dynamic approach than signature-based detection, it also requires accurate profiling of what is considered regular network behavior. However, like the flaws of other conventional intrusion detection methods, its effectiveness is greatly affected by the nature of constantly evolving cyber threats.

As we embark on a new era of artificial intelligence in cyberspace, modern intrusion detection systems must leverage machine learning-based approaches [7]. These techniques enable systems to autonomously learn patterns in network data, enhancing their capability to identify anomalies and potential security breaches. Supervised learning utilizes labeled datasets to train models in distinguishing normal and malicious behavior. Unsupervised learning detects anomalies without labeled data, while semi-supervised learning combines both approaches. Such paradigms empower intrusion detection systems to dynamically adapt to evolving threats, offering proactive defense mechanisms for safeguarding critical network infrastructure. Utilizing advanced modern machine learning methods, we employ a deep neural network with a final layer comprising 4 neurons for 4-class classification, facilitating the categorization of network activities into distinct classes. To address the challenge of heavily imbalanced datasets, we employ Conditional Tabular Generative Adversarial Network (CTGAN) sampling, a method tailored for generating synthetic samples to balance the dataset effectively [8]. Overall, in summary, the major contributions of this paper consist of the

The Department of Homeland Security (DHS), grant awards 21STSLA00011-01-0 and 23STSLA00017-01-00, is gratefully acknowledged by the authors.

following key points:

- Introduced a novel method for effectively handling imbalanced datasets like NSL-KDD [9], boosting the robustness of intrusion detection systems.
- Demonstrated deep neural networks (DNNs) feature compression capability and its integration with CTGAN oversampling for improved efficiency in managing large datasets.
- Outperforms traditional algorithms [7], [10] by utilizing minimal features at the final layer of the neural network, making it much more efficient.

This paper comprises five sections. A concise introduction is provided in Section I. Section II briefly describes the use of both classical machine learning models and deep learning model applications in previous relevant studies. In Section III, we describe the neural network, training, proposed scheme, and how it works well in conjunction with CTGAN for minority oversampling. Section IV presents the performance and comparative analysis for network intrusion detection. Finally, Section V describes the conclusion and the future indication of the research.

II. Literature Review

This section describes the applications of traditional machine learning algorithms, ensemble-based techniques, and deep learning algorithms for intrusion detection. Primarily the artificial neural networks along with their unique utilization in the field of network intrusion detection will be reviewed.

Traditional machine learning algorithms have been employed in recent studies to detect and classify network anomalies in widely used operating systems like Windows and Linux. These models leverage classical ML algorithms such as K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Naïve Bayes, Decision Tree, and Logistic Regression [11], [12]. While traditional machine learning approaches have historically proven effective against various cyber threats, each algorithm possesses its weaknesses. Ensemble-based algorithms mitigate individual weaknesses by combining multiple weaker classifiers to form a robust model. Commonly used ensemble methods in machine learning include Random Forest, Gradient Boosting, and Stacking [13], [14]. For instance, paper [14] demonstrated the efficacy of a Stacking ensemble-based approach in network intrusion detection. This model adopts a collective decision-making strategy, aggregating votes from diverse classifiers to produce comprehensive and accurate classifications.

Recent studies [15], [16] demonstrate the effectiveness of deep learning in various cyber intelligence areas, including malware and network intrusion analysis. Deep Neural Networks (DNN) are not confined to strict flexibility, their architecture can vary greatly depending on both their field of research and their specific use case. Neural network architectures such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Auto-encoders (AEs) have all had vital roles in contemporary intrusion detection systems [17], [18]. For instance, paper [18] introduced an AE capable of real-time feature learning and dimensionality reduction, beneficial for both network-based intrusion detection and malware classification. Similarly, paper [19] showcased a CNN based on VGG-16 architecture achieving commendable performance in network intrusion classification. Additionally, paper [20] utilized sparse autoencoder (SAE) for NSL-KDD data analysis, demonstrating the benefits of deep learning approaches.

The proposed model leverages the capabilities of deep learning for oversampling, dimensionality reduction, and multi-class classification, eliminating the need for feature selection methods. By leveraging CTGAN to handle class imbalance and employing a deep neural network, our approach achieves both dimensionality reduction and 4-class classification of network anomalies.

III. Methodology

This paper involved the implementation of a specialized architecture known as Conditional Tabular Generative Adversarial Network (CTGAN) [8]. This unique model is specifically designed to handle tabular data, making it effective for generating robust synthetic data to handle class imbalance. The well-suited DNN for large datasets is utilized for both training and 4-class classification with SoftMax Regression. The following sections will discuss the different aspects of the proposed model in detail.

A. Dataset Description

This paper utilizes the well-established NSL-KDD dataset, an enhanced version of the KDD-Cup'99 dataset [9]. The information-rich dataset consists of 43 total features, comprising different data types such as categorical, integer, binary, and real number values to ensure robustness. Samples are classified into normal and attack classes. The attack class is further divided into DoS, Probe, R2L, and U2R categories. The distribution of these categories is illustrated in Table I.

TABLE I
Training and Test Sample distribution of NSL-KDD dataset

Class	Training Sample	Test Sample
DoS	45927	7458
Probe	11656	2421
R2L	995	2887
U2R	52	67
Total	58630	12833

B. Data Preprocessing

Data preprocessing is essential for constructing a robust machine-learning model. This includes label encoding for categorical features, min-max normalization for integer features, and establishing the target class distribution for our 4-class classification model. Furthermore, our proposed model incorporates additional techniques like data cleaning and feature elimination. The following subsections describe each data preprocessing step outlined in this article.

TABLE II
INTRUSION CLASS DISTRIBUTION IN TRAINING SET

Intrusion Label	Intrusion Type
DoS	Neptune, Smurf, Back, Teardrop, Pod, Land
Probe	Satan, Ipsweep, Portsweep, Nmap
U2R	Warezclient, Guess_Passwd, Warezmaster, Imap, Ftp_Write, Multihop, Phf, Spy
R2L	Buffer_Overflow, Rootkit, Loadmodule, Perl

TABLE III
INTRUSION CLASS DISTRIBUTION IN TEST SET

Intrusion Label	Intrusion Type
DoS	Neptune, Smurf, Apache2, Processtable, Back, Mailbomb, Teardrop, Pod, Land, Udpstorm
Probe	Mscan, Satan, Saint, Ipsweep, Portsweep, Nmap
U2R	Guess_Passwd, Ftp_Write, Warezmaster, Snmpguess, Snmpgetattack, Httptunnel, Multihop, Named, Sendmail, Xlock, Xsnoop, Imap, Phf, Worm
R2L	Buffer_Overflow, Ps, Rootkit, Xterm, Loadmodule, Perl, Sqlattack

1) Label Encoding: We utilized label encoding to convert our categorical data into a machine-readable format, assigning integer values starting from '0', '1', '2', and so on. Once the class distribution was established, it was then applied to the target variable.

2) Data Cleaning: While analyzing the dataset, it was discovered that one of the binary features, 'su_attempted', had a value of '2'. This feature was subsequently converted to '1' to improve model learning and ensure overall consistency. As the other features had already been cleaned, no further data cleaning was necessary for them.

3) Feature Elimination: Upon employing statistical measures, it was observed that the integer feature 'num_outbounds_cmds' had a consistent value of '0' across all records. Features lacking significance to the target class should be eliminated to enhance model performance. Therefore, it was decided to drop this feature.

4) Min-Max Normalization: To ensure consistency in the dataset, min-max normalization was applied to our numeric integer features using equation 1. This process scales each value to fall within the range of [0,1]. This normalization aids in model training. Since real (continuous) features present in the NSL-KDD dataset were already normalized to fall in the range of [0,1], no further modifications were required on them.

$$X_{norm} = \frac{(X - X_{min})}{(X_{max} - X_{min})} \tag{1}$$

5) Intrusion Class Distribution: The NSL-KDD dataset contains numerous intrusion types. Although the dataset includes many benign samples, the 'normal' class was removed, to perform a 4-class classification of the distinct intrusion labels. Tables II and III present a comprehensive list of these intrusions and their corresponding broader classes, including DoS, Probe, U2R, and R2L.

C. Synthetic Data Generation using CTGAN

To address the substantial class imbalance within the dataset, we utilized CTGAN to generate synthetic samples. With a batch size of 5000 with epochs at 100, the model underwent 100 training iterations for comprehensive dataset exposure. A learning rate of 0.0002 ensured the balanced adjustment. Regularization was applied with beta_1 and beta_2 set to 0.5 and 0.9, controlling exponential decay rates in the Adam optimizer, and stabilizing the training process. The intensive training phase of the GAN ensures that the generator can produce synthetic datasets closely resembling real-world data characteristics. Although a total of 300,000 synthetic samples were generated, those related to 'DoS' and 'normal' were excluded due to the irrelevance of the 4-class imbalance. The remaining synthetic samples for 'Probe', 'R2L', and 'U2R' were combined with the original training samples to create a balanced dataset. The combined distributions of the training data samples are delineated in Table IV.

TABLE IV
TRAINING SAMPLE DISTRIBUTION AFTER APPLYING CTGAN FOR MINORITY CLASSES

Class	Sample Size
DoS	45927
Probe	40974
R2L	4061
U2R	302
Total	91264

D. Classification with Deep Neural Network

As previously discussed, deep neural networks (DNNs) excel in modern machine learning, proficiently learning patterns from large datasets and offering exceptional classification capabilities. Figure 1 illustrates the architecture of our proposed DNN model, comprising an input layer, 3 hidden layers with 80, 40, and 4 neurons, and an output layer with Softmax activation for 4-class classification. Neurons initially double in each layer and then progressively compress until reaching 4 neurons, supporting dimensionality reduction and efficient training. Rectified Linear Unit (ReLU) activation with 'categorical_crossentropy' as the loss function is used in the hidden layer. The network is trained using batch learning where the batch size is set to 32. To validate the model in each iteration, 15% of the training data is set aside. Further, the learning rate of the network is set to 0.015 and Adam

optimizer is used to minimize the loss function during the training of the network. To reduce model bias, early stopping is implemented to stop the training of the network at iteration six.

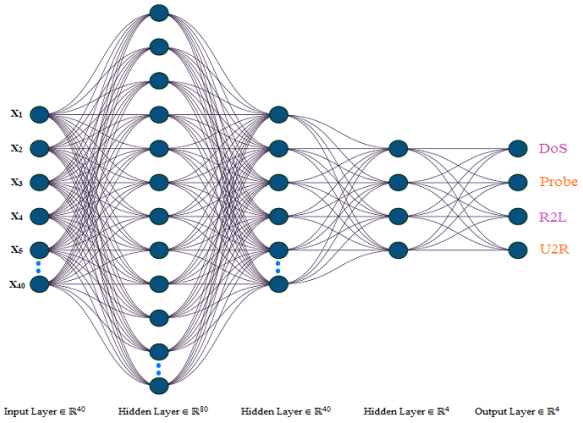

Fig. 1. Architecture of the Deep Neural Network.

E. Proposed Model

Figure 2 presents a detailed top-level view of the model topology in the form of a block diagram. The process starts with data collection, followed by sequence of stages involving data analysis, preprocessing, dataset preparation, and ends with classification of intrusion samples. By combining the use of well-thought-out data handling, CTGAN for synthetic data generation, and DNNs for performing classification, the proposed framework is able to achieve an outstanding performance. To ensure the reproducibility of the proposed work, the associated dataset and code are made available on GitHub.

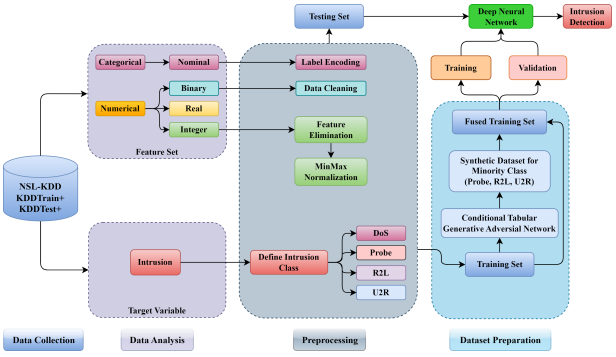

Fig. 2. The detailed block diagram of the proposed model.

F. Experimental Setup

Experiments were rigorously performed on a high-performance computing system featuring an Intel Core i7 9750H six-core CPU with 16 GB RAM. Essential Python libraries included scikit-learn (v1.2.2) for model training and evaluation, matplotlib (v3.7.4) for insightful visualizations of performance metrics, and pandas (v2.0.3) for seamless data manipulation and analysis were used. Keras (v2.12.0) with TensorFlow 2.12.0 backend was used to construct intricate neural networks to classify network intrusions precisely. Additionally, we integrated the ydata-synthetic library (v1.3.1) for CTGAN synthetic data generation, enhancing model robustness against the imbalanced dataset.

IV. Result Analysis

This section discusses the performance analysis of the proposed model along with illustrative images to provide visualization. The performance of the proposed model is evaluated using metrics such as precision, recall, f1-score, accuracy, and Matthew's correlation coefficient (MCC) derived from the concatenation of the real NSL-KDD data with the newly generated synthetic data from the CTGAN. To establish the effectiveness of the proposed model, it is further compared with the state-of-the-art methods.

A. Performance Evaluation Metrics

Precision quantifies the proportion of correctly predicted positive instances relative to all predicted positives, offering insights into the accuracy of positive predictions. Recall, on the other hand, measures the percentage of actual positive instances correctly identified by the model. The F1-score, serving as the harmonic mean of precision and recall, provides a balanced evaluation of a model's effectiveness. Accuracy reflects the overall correctness of predictions, while the Matthews Correlation Coefficient (MCC) combines true positives, true negatives, false positives, and false negatives to evaluate classification performance, particularly in imbalanced datasets. Initially, class-wise precision and recall are computed using equations 2 and 3. Subsequently, weighted precision and recall are determined via equations 4 and 5. Finally, the weighted F1 score, accuracy, and MCC are calculated using equations 6, 7, and 8,

$$Precision_i = \frac{TP_i}{(TP_i + FP_i)} \tag{2}$$

$$Recall_i = \frac{TP_i}{(TP_i + FN_i)} \tag{3}$$

$$Precision_{weighted} = \frac{\sum_{i=0}^{3} Precision_i \times t_i}{Total\ Samples} \tag{4}$$

$$Recall_{weighted} = \frac{\sum_{i=0}^{3} Recall_i \times t_i}{Total\ Samples} \tag{5}$$

$$F1\ Score_{weighted} = 2 \times \frac{Precision_{weighted} \times Recall_{weighted}}{Precision_{weighted} + Recall_{weighted}} \tag{6}$$

$$Accuracy = \frac{\sum_{i=0}^{3} TP_i + \sum_{i=0}^{3} TN_i}{Total\ Samples} \tag{7}$$

$$MCC = \frac{c \times s - \sum_{i=0}^{3} p_i \times t_i}{\sqrt{(s^2 - \sum_{i=0}^{3} p_i^2) \times (s^2 - \sum_{i=0}^{3} t_i^2)}} \tag{8}$$

Where, i = class index; s = total samples; c = correctly predicted samples; t_i = number of samples in class i; p_i = number of samples predicted as class i; TP_i = number of samples correctly predicted as class i; TN_i = number of samples correctly predicted as not class i; FP_i = number of samples falsely predicted as class i; FN_i = number of samples falsely predicted as not class i;

B. Performance Evaluation

This section is divided into three subsections: training the DNN, evaluating the proposed model's performance on the independent test dataset, and conducting a comparative analysis with the state-of-the-art studies employing GAN technology.

1) Training Deep Learning Model: The neural network underwent extensive training across multiple distinct layers utilizing the Adam optimizer, with accuracy as the primary monitored metric. Figure 3 displays a plot generated from the trained model, showcasing both the training accuracy (train_accuracy) and validation accuracy (val_accuracy). The graph illustrates a training accuracy of 97.8% and a 10-fold cross-validation accuracy of 93.06%.

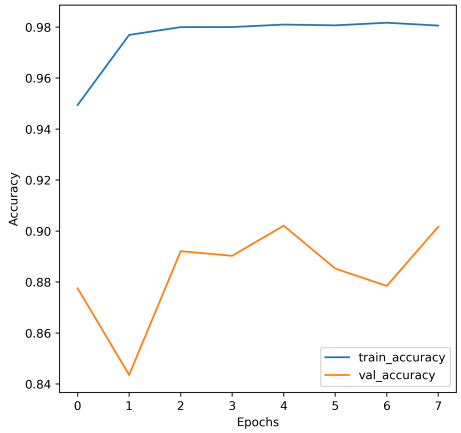

Fig. 3. Training and validation accuracy during the model building.

2) Performance analysis on test dataset: A confusion matrix heatmap was generated using the test dataset, depicted in Figure 4 to thoroughly evaluate the model's performance. Rows represent the actual classes, while columns display the predicted classes generated by the proposed model. Each cell contains values, with darker colors highlighting a higher percentage for a particular class. For instance, among 7,458 samples, 6,472 were accurately classified as class DoS, and out of 67 samples, 30 were correctly identified as class U2R. Additionally, an essential assessment of the model's performance is illustrated through Receiver Operating Characteristic (ROC) curves for each class, presented in Figure 5. However, when examining the ROC curve, it's inadequate for comparing classifier performances alone. A scalar value is necessary to determine the efficacy of the classifier. Hence, the Area under the ROC curve (AUC) serves this purpose, indicating the model's effectiveness. A higher AUC value correlates with better performance. Plotting the ROC curve is challenging compared to binary classification in the context of multi-class classification tasks. To address this, we designate a specific class as positive samples while treating the rest as negative samples to generate ROC curves for each class. Remarkably, all classes exhibit exceptional AUC values on the NSL-KDD dataset. Notably, the R2L attack class attains an impressive 97% AUC value.

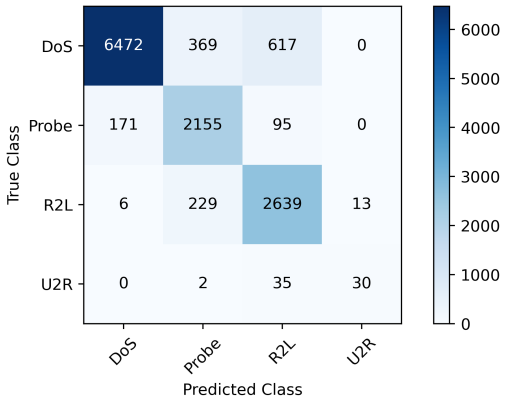

Fig. 4. Confusion matrix of 4 intrusion classes on the test dataset.

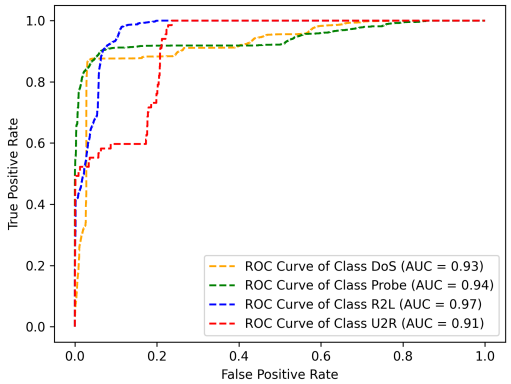

Fig. 5. ROC curve of 4 intrusion classes on the test dataset.

3) Comparative Analysis: The proposed model is evaluated against the state-of-the-art DGM [7], which employs CGAN and is assessed on the NSL-KDD and UNSW-NB15 datasets. DGM examines performance metrics for each attack class separately. To facilitate comparison with our proposed model, the independent performance metrics of DGM's attack classes are consolidated to signify the comprehensive performance of each DGM model, calculated as the average of class-wise precision, recall, and F1 score. The precision, recall, and F1 score presented in Table V for DGM correspond to these average values. Additionally, another study [10] introduces WCGAN with a gradient penalty objective function to enhance model stability and minimize loss. Upon reviewing Table V, it is evident that our proposed model outperforms both DGM and WCGAN models. Precision, recall, and F1 score have seen substantial enhancements of 105.93%, 56.37%, and 80.81%, respectively. This improvement is attributed to the integration

TABLE V
COMPARATIVE PERFORMANCE ANALYSIS OF PROPOSED MODEL ON DIFFERENT DATA SYNTHESIS APPROACH AND ML CLASSIFIERS

Model	Synthesis Approach	Classifier	Precision	Recall	F1 Score	Accuracy	MCC
DGM [7]	CGAN	Random Forest (imp. %)	63.50 (40.50%)	54.50 (61.50%)	58.80 (50.71%)	-	-
		Decision Tree (imp. %)	51.75 (72.41%)	53.54 (64.40%)	52.63 (68.38%)	-	-
		Neural Network (imp. %)	60.50 (47.47%)	55.50 (58.59%)	57.89 (53.08%)	-	-
		SVM (imp. %)	61.00 (46.26%)	54.00 (63.00%)	57.29 (54.69%)	-	-
XGBoost-WCGAN [10]	WCGAN	Random Forest (imp. %)	83.32 (7.08%)	72.30 (21.74%)	77.42 (14.47%)	-	-
		Decision Tree (imp. %)	69.24 (28.86%)	68.80 (27.94%)	69.02 (28.40%)	-	-
		XGBoost (imp. %)	86.70 (2.91%)	88.47 (-0.51%)	88.58 (0.045%)	-	-
		SVM (imp. %)	12.71 (601.97%)	34.62 (154.3%)	18.59 (376.7%)	-	-
DNN-CTGAN	**CTGAN**	**DNN (avg. imp. %)**	**89.22 (105.93%)**	**88.02 (56.37%)**	**88.62 (80.81%)**	**88.02 (-)**	**80.47 (-)**

Note: Here, 'imp. %' represents the percentage enhancement by DNN-CTGAN over the respective method, while 'avg. imp. %' denotes the average enhancement across listed methods.

of CTGAN with a deep neural network for data synthesis, feature reduction, and classification.

V. CONCLUSIONS

In this work, we present a novel method for effectively addressing imbalances in the NSL-KDD dataset and enhancing the robustness of intrusion detection systems. Initially, we employ the CTGAN oversampling technique to generate synthetic samples, focusing on balancing the dataset's minority classes. Subsequently, we utilize a deep neural network (DNN) for feature compression and classification, aiming to improve efficiency in handling large datasets. We also evaluated the effectiveness of the balanced dataset, produced using a generative model, in multi-class classification using precision, recall, F1 score, accuracy, and MCC as evaluation metrics. The proposed approach resulted in encouraging outcomes in classifying the samples of DoS, Probe, R2L, and U2R attacks. Specifically, the proposed method results in an average improvement of 105.93%, 56.37%, and 80.81% based on precision, recall, and F1 score compared to the state-of-the-art methods. Nevertheless, certain shortcomings remain, particularly regarding minority class representation with the CTGAN oversampling technique. In the future, we propose exploring more intricate learning techniques to address data imbalance and enhance generalizability.

REFERENCES

[1] O. E. Aeraj and C. Leghris, "Study of the SNORT intrusion detection system based on machine learning," *in 2023 7th IEEE Congress on Information Science and Technology (CiSt)*, 2023, pp. 33-37.

[2] D. S. P. Puvvala, G. Madala, M. Kada and U. Hariharan, "Improved Network Intrusion Detection System Using Deep Learning," *in 2023 7th International Conference on Electronics, Materials Engineering & Nano-Technology (IEMENTech)*, 2023, pp. 1-6.

[3] H. Hindy et al., "A Taxonomy of Network Threats and the Effect of Current Datasets on Intrusion Detection Systems," *IEEE Access*, vol. 8, pp. 104650-104675, 2020.

[4] A. K. Balyan et al., "A hybrid intrusion detection model using ega-pso and improved random forest method," *Sensors*, 22(16), p.5986, 2022.

[5] S. Wang, C. Xia and T. Wang, "A Novel Intrusion Detector Based on Deep Learning Hybrid Methods," *in 2019 IEEE 5th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS)*, 2019, pp. 300-305.

[6] R. Kumar and D. Sharma, "HyINT: Signature-Anomaly Intrusion Detection System," *in 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, 2018, pp. 1-7.

[7] G. Dlamini, and M. Fahim, "DGM: a data generative model to improve minority class presence in anomaly detection domain," *Neural Computing and Applications*, 33, pp.13635-13646, 2021.

[8] L. Xu, M. Skoularidou, A. Cuesta-Infante, K. Veeramachaneni. "Modeling Tabular data using Conditional GAN," *Advances in neural information processing systems 32 (2019).*

[9] M. Tavallaee, E. Bagheri, W. Lu, and A. Ghorbani, "A Detailed Analysis of the KDD CUP 99 Data Set," *in 2009 IEEE symposium on computational intelligence for security and defense applications*, 2009, pp. 1-6.

[10] V. Kumar, and D. Sinha, "Synthetic attack data generation model applying generative adversarial network for intrusion detection," *Computers and Security*, 125, p.103054, 2023.

[11] C. Chen, L. Song, C. Bo and W. Shuo, "A Support Vector Machine with Particle Swarm Optimization Grey Wolf Optimizer for Network Intrusion Detection," *in 2021 International Conference on Big Data Analysis and Computer Science (BDACS)*, 2021, pp. 199-204.

[12] F. Chen, Z. Ye, C. Wang, L. Yan and R. Wang, "A Feature Selection Approach for Network Intrusion Detection Based on Tree-Seed Algorithm and K-Nearest Neighbor," *in 2018 IEEE 4th International Symposium on Wireless Systems within the International Conferences on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS-SWS)*, 2018, pp. 68-72.

[13] R. A. Disha and S. Waheed, "A Comparative study of machine learning models for Network Intrusion Detection System using UNSW-NB 15 dataset," *in 2021 International Conference on Electronics, Communications and Information Technology (ICECIT)*, 2021, pp. 1-5.

[14] M. Vishwakarma, and N. Kesswani, "A new two-phase intrusion detection system with Naïve Bayes machine learning for data classification and elliptic envelop method for anomaly detection," *Decision Analytics Journal*, 7, p.100233, 2023.

[15] J. A. Abraham and V. R. Bindu, "Intrusion Detection and Prevention in Networks Using Machine Learning and Deep Learning Approaches: A Review," *in 2021 International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA)*, 2021, pp. 1-4.

[16] Y. Dong, R. Wang and J. He, "Real-Time Network Intrusion Detection System Based on Deep Learning," *in 2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS)*, 2019, pp. 1-4.

[17] S. M. Kasongo, "A deep learning technique for intrusion detection system using a Recurrent Neural Networks based framework," *Computer Communications*, 199, pp.113-125, 2023.

[18] M. Yousefi-Azar, V. Varadharajan, L. Hamey and U. Tupakula, "Autoencoder-based feature learning for cyber security applications," *in 2017 International Joint Conference on Neural Networks (IJCNN)*, 2017, pp. 3854-3861.

[19] M. Masum and H. Shahriar, "TL-NID: Deep Neural Network with Transfer Learning for Network Intrusion Detection," *in 2020 15th International Conference for Internet Technology and Secured Transactions (ICITST)*, 2020, pp. 1-7.

[20] K. N. Rao, K. V. Rao, and P. R. PVGD, "A hybrid intrusion detection system based on sparse autoencoder and deep neural network," *Computer Communications*, 180, pp.77-88, 2021.