Enhanced and Explainable Deep Learning-Based Intrusion Detection in IoT Networks

Sohan Gyawali*, Kamran Sartipi†, Benjamin Van Ravesteyn*, Jiaqi Huang‡, Yili Jiang§
*Department of Technology Systems, East Carolina University, NC, USA, Email: gyawalis22@ecu.edu
†Department of Computer Science, East Carolina University, NC, USA, Email: sartipik16@ecu.edu
‡Department of Computer Science and Cybersecurity, University of Central Missouri, MO, USA, Email: jhuang@ucmo.edu
§Department of Computer and Information Science, University of Mississippi, MS, USA, Email: yjiang7@olemiss.edu

***Abstract*—The proliferation of IoT networks has significantly increased the potential for cyber attacks. Deep learning models have shown effectiveness in detecting complex attacks; however, they face challenges related to imbalanced datasets and a lack of interpretability. In this work, we propose an enhanced and interpretable deep learning approach that addresses the common challenges of data imbalance and interoperability. To tackle the data imbalance issue, we employ CTGAN, a technique that expands the dataset by generating synthetic samples for the minority class traffic. Additionally, we utilize Boruta Shap for feature extraction, resulting in a reduced number of features and enhancing the efficiency of detection. Moreover, we incorporate SHAP for model explainability. We validate the results obtained from SHAP by conducting a thorough analysis of each attack type in both the NSL-KDD and UNSW-NB15 datasets. Furthermore, we conduct a comparative evaluation of our framework against a previous approach, demonstrating that our proposed framework outperforms the previous one in accurately detecting attacks for majority of class types.**

***Index Terms*—Explainable intrusion detection, deep learning, IoT security**

I. Introduction

The Internet of Things (IoT) is an evolving communication technology that gathers information from diverse sources and provides remote functionality [1]. Recently, IoT devices and sensors have witnessed significant growth across several areas such as intelligent transportation systems, healthcare systems, and smart homes. Despite the numerous advantages offered by these IoT systems, minimal consideration is given to security during the design and development phase by manufacturers to meet the market demands [2].

Intrusion detection plays a crucial role in strengthening security within IoT networks. By implementing intrusion detection systems (IDS), organizations can proactively monitor and analyze network traffic to identify potential security breaches or malicious activities. Recently, there has been a significant increase in the adoption of machine learning-based IDS in IoT networks, both in industry and academia [3]–[7]. Machine learning or deep learning-based IDS are trained on a dataset of normal and malicious behaviors. After training, these systems can learn to detect anomalies or potential intrusions automatically. Machine learning-based detection can adapt and improve over time as new threats emerge, making it a valuable approach for IoT security. While notable advancements have been made in intrusion detection through the use of AI-powered techniques, there are still obstacles when it comes to deploying them in real-world systems.

One of the primary obstacles faced in deep learning-based intrusion detection in IoT is the issue of data imbalance. Typically, the majority of network flow data consists of normal traffic, while instances of malicious behavior are relatively rare. Moreover, most of the available data pertains to well-known attacks such as Denial of service (DoS) and Probe, while other specific attacks are extremely infrequent. As a result, deep learning-based IDS may struggle to sufficiently learn the characteristics of specific network threats. Additionally, deep learning models are inherently non-transparent, making it challenging to understand the reasoning behind their decisions.

To address the aforementioned challenges, we introduced an innovative deep learning-based intrusion detection system for IoT networks. Our proposed framework effectively tackled both the data imbalance and explainability issues. We accomplished this by employing conditional generative adversarial networks (CTGANs) [8] to generate realistic synthetic network traffic data, specifically targeting minor attack traffic instances. Additionally, the paper incorporated the SHapley Additive exPlanations (SHAP) [9] mechanism to enhance the transparency and resilience of deep learning-based IDS in IoT networks. Our framework not only enabled the interpretation of decisions made by the deep learning-based IDS but also facilitated feature selection. This feature selection capability reduces performance costs while maintaining high detection accuracy.

The rest of this paper is organized as follows. Section II summarises various related works. Section III discusses the proposed framework. Section IV presents the experiments and results. Finally, Section V concludes the paper.

II. Related work

Several studies have examined the utilization of deep learning methods in IDS for IoT networks or networks in general.

In the paper [10], a deep learning approach based on bidirectional long short-term memory (LSTM) architecture was utilized for intrusion detection in the context of the Internet of Vehicles (IoV). The proposed framework was evaluated using the UNSW-NB15 dataset [11] and car hacking data source. Impressive results were obtained, with the framework achieving a high accuracy rate of 98.88% for the UNSW-NB15

dataset. Similarly, the authors in [12] investigated a detection system that combined autoencoders and LSTM for intrusion detection. Autoencoders were employed for feature extraction, and LSTM was utilized for the detection process. The proposed system achieved a detection accuracy of over 92% when evaluated on the UNSW-NB15 dataset. In a different study [13], the authors proposed an intrusion detection system based on a stacked autoencoder (SAE) and a deep neural network (DNN). The system's performance was evaluated using the NSL-KDD dataset [14], and it achieved high accuracy for multiclass classification tasks. The aforementioned papers, however, do not address the challenges associated with imbalanced datasets and the lack of feature explainability in deep neural networks.

In the paper [15], the authors addressed the issue of interpretability in deep learning-based IDS by employing three explainable artificial intelligence (XAI) techniques: RuleFit, SHAP, and local interpretable model-agnostic explanations (LIME). These techniques were used to provide both local and global explanations, improving the interpretation of decisions made by deep learning-based IDS. The proposed method was validated using the NSL-KDD and UNSW-NB15 datasets. Similarly, in [16], the authors proposed an explainable deep learning-based intrusion detection framework aimed at enhancing transparency and resilience in deep learning-based IDS for IoT networks. In their work, the authors employed the SHAP technique to interpret the decisions made by the deep learning-based IDS. The proposed framework was evaluated using the ToN_IoT dataset, achieving high performance in terms of intrusion detection. However, both of the aforementioned papers employed XAI techniques without addressing the issue of imbalanced datasets. This can pose a challenge as attacks occurring infrequently in the datasets may not be well explained by the IDS due to the data imbalance.

In [17], the authors utilized a cutting-edge generative model to generate synthetic data specifically targeting minor attack traffic. They employed a generative adversarial network (GAN) architecture based on the Wasserstein distance and combined it with autoencoder-driven deep learning models. Through comprehensive evaluations, the authors demonstrated that their proposed scheme outperformed previous AI-based IDS approaches in terms of performance. Similarly, in [18], the authors tackled the issue of data imbalance by combining a conditional Wasserstein generative adversarial network (CWGAN) with cost-sensitive stacked autoencoders (CASSAE). This approach aimed to improve the detection accuracy of minority and unknown attacks in the NSL-KDD and UNSW-NB15 datasets. The authors showcased the effectiveness of their scheme in enhancing detection performance. Both the aforementioned works utilized the Wasserstein distance-based GAN, which may not be suitable for handling heterogeneous data. Additionally, neither of the studies addressed the lack of explainability in their models, which is an important aspect of understanding and interpreting the decisions made by IDS.

In contrast to previous studies, our proposed framework jointly addressed the challenges of data imbalance and explainability. We achieved this by utilizing CTGANs to generate authentic synthetic network traffic data, focusing specifically on generating realistic instances of minor attack traffic. Moreover, our approach integrated the SHAP mechanism to improve the transparency and robustness of deep learning-based IDS.

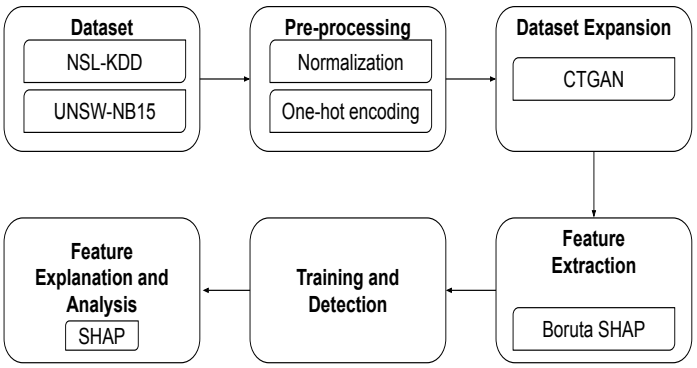

Fig. 1. Proposed Framework

III. Proposed Framework

The framework we propose, as shown in Fig. 1, comprises five main stages: 1) Pre-processing, 2) Dataset expansion, 3) Feature extraction, 4) Training and detection, and 5) Feature explanation and analysis.

A. Pre-processing

Pre-processing primarily involves two steps: standardization and one-hot encoding. Standardization is applied to numerical features which involve scaling using a standardized method. Categorical or nominal features are subjected to one-hot encoding to represent them in a binary vector.

B. Dataset expansion

In general, most of the data flowing through a network is regular traffic, and instances of malicious activity are quite infrequent. Consequently, deep learning-based systems designed to detect intrusions may encounter difficulties in effectively understanding the unique features of specific network threats. To address this issue, we utilized CTGANs [8].

CTGAN is based on Generative Adversarial Networks (GANs) and aims to model the distribution of tabular data and generate sample rows from that distribution. It tackles various challenges by utilizing a conditional generator, which effectively models both discrete and continuous columns. CTGAN introduces innovative features, including mode-specific normalization during training, where continuous feature values are represented by a one-hot vector indicating the sampled mode and a normalized scalar value. The conditional generator addresses issues caused by imbalanced categories, which often result in mode collapse in GANs. However, conditional architectures come with limitations: the input must be properly prepared for the generator to interpret the conditions, and the generated rows must adhere to the input conditions.

There are other GAN-based architectures such as WGAN [19] and WGAN-GP [20] that attains stability in terms of training. However, they encounter challenges when it comes to handling mixed data types. In contrast, CTGAN was specifically developed to tackle the difficulties posed by tabular datasets that contain a combination of numeric and categorical features. In our study, we utilized CTGAN to generate realistic synthetic data that mimics network traffic. Our main focus was specifically on creating instances of minor attack traffic as shown in Algorithm 1.

Algorithm 1: CTGAN-based Dataset Expansion

Input: Initial IDS Dataset - $\mathcal{D}_{\mathcal{I}}$
Output: Expanded IDS Dataset -$\mathcal{D}_{\mathcal{E}}$
Identify: $\mathcal{K}$ minority attack class in $\mathcal{D}_{\mathcal{I}}$
Initialize: CTGAN generators, $\mathcal{G}$
Initialize: $\hat{\mathcal{D}_{\mathcal{J}}} = [\,]$; `// empty list`
foreach $\mathcal{J} \in \mathcal{K}$ **do**
  $\hat{\mathcal{T}_{\mathcal{J}}} = \mathcal{G}_{\mathcal{J}}(\mathcal{D}_{\mathcal{J}})$
  $\hat{\mathcal{D}_{\mathcal{J}}} = \hat{\mathcal{D}_{\mathcal{J}}} + \hat{\mathcal{T}_{\mathcal{J}}}$
end
$\mathcal{D}_{\mathcal{E}} = \mathcal{D}_{\mathcal{I}} \cup \hat{\mathcal{D}_{\mathcal{J}}}$

C. Features extraction

Dealing with a large number of features in intrusion detection can pose challenges such as increased memory requirements, higher processing power demands, and large performance overhead. To address this, Boruta [21], a feature selection algorithm, can be used which leverages two principles: Shadow Features and Binomial Distribution.

In Boruta, features are not pitted against each other directly. Instead, they are compared to their randomized counterparts known as shadow features. The goal of Boruta is to identify features that outperform shadow features. If a feature exhibits greater importance than a predefined threshold, it is considered a 'hit'. Every feature is categorized as either a hit or not a hit, enabling the formation of a binomial distribution from these outcomes. Although Boruta is a powerful technique for selecting relevant features, its effectiveness depends on accurately calculating feature importances, which can be influenced by inadequate data. To overcome this, Boruta SHAP [22], a feature selection algorithm can be used which combines Boruta with SHAP Values. By incorporating SHAP values into Boruta, we gain the ability to obtain comprehensive feature explanations offered by SHAP, while still benefiting from the robustness of the Boruta algorithm, ensuring that only significant variables are retained in the feature set.

D. Training and Detection

After completing the feature selection procedure, we employed a deep neural network (DNN) to train and detect patterns. DNNs have demonstrated their effectiveness in identifying intricate attacks, which is why we opted for this model. Our framework also allows for the utilization of other forms of deep learning, such as convolutional neural networks and autoencoders. Nevertheless, the main focus of this research paper is to jointly tackle the challenges of data imbalance and explainability.

E. Feature explanation and analysis

To provide explanations for the features in our intrusion detection system, we have utilized SHAP [9]. SHAP is a framework designed to interpret the relevance of features by assigning importance values specific to each prediction. It is built upon the mathematical concept of Shapley values derived from cooperative game theory. Deep SHAP is an approximation algorithm specifically developed to compute SHAP values for deep learning models. It capitalizes on the connection between DeepLIFT [23], and Shapley values. This connection allows Deep SHAP to efficiently estimate SHAP values in deep neural networks.

The primary aim of Deep SHAP in our framework is to interpret the prediction made by our system by determining the contribution of each feature to that prediction. This entails assessing the impact of each feature on the overall prediction generated by our intrusion detection system.

IV. Experiments and Results

In our study, we utilized two widely recognized datasets, namely NSL-KDD [14] and UNSW-NB15 [11], which are commonly employed as benchmark datasets for intrusion detection. The primary motivation behind selecting these datasets was to facilitate result comparison with other research in the field.

A. Datasets

The NSL-KDD dataset is extensively used in intrusion detection studies for IoT systems. It comprises 125,973 instances in the KDDTrain subset and 22,544 instances in the KDDTest subset. This dataset contains 41 distinct features that represent various aspects of network flow, alongside a label indicating whether the instance is categorized as normal or an attack. The attacks are classified into four main types: DoS, Probing, remote to local (R2L), and user to root (U2R).

Similarly, the UNSW-NB15 dataset is another prominent dataset used in the field of intrusion detection for IoT systems. The training dataset of UNSW-NB15 consists of 175,341 observations, while the testing dataset comprises 82,332 observations. It contains 43 features and two class features that indicate whether an instance is classified as normal or an attack. The dataset covers nine primary attack types: Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms.

B. Implementation

To prepare both the NSL-KDD and UNSW-NB15 datasets for analysis, we conducted preprocessing tasks, which primarily consisted of standardization and one-hot encoding. These steps ensured the data was appropriately formatted and ready for further analysis.

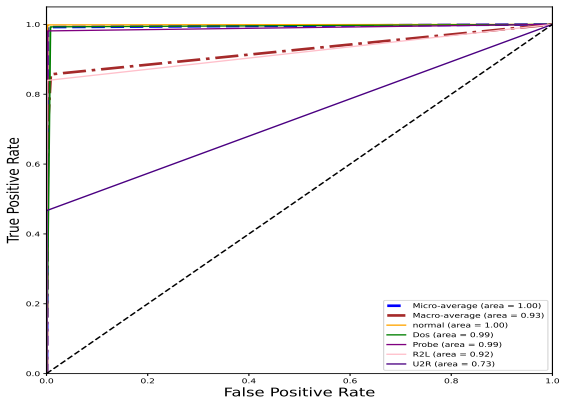

Fig. 2. ROC curve without dataset expansion and feature extraction for NSL-KDD

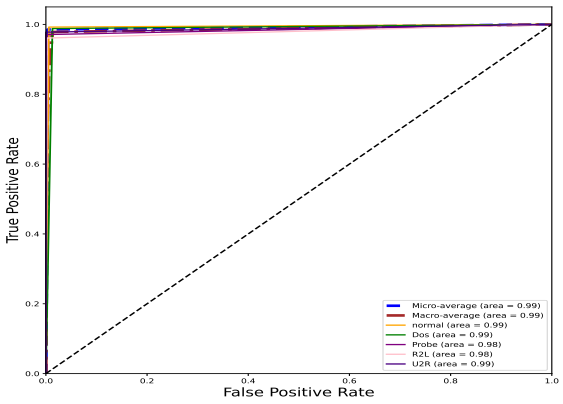

Fig. 3. ROC curve after dataset expansion and feature extraction for NSL-KDD

Following preprocessing, we utilized CTGAN to augment the dataset. In the case of the NSL-KDD dataset, we specifically expanded the dataset for the less common attack types, namely R2L and U2R. We introduced an additional 10,000 rows for each of these attack types. Similarly, for the UNSW-NB15 datasets, we included 5,000 new rows for each of the minority attack types, which include Analysis, Backdoors, Shellcode, and Worms.

After enlarging the dataset, we employed Boruta Shap to select the most important features for both the NSL-KDD and UNSW-NB15 datasets. For the NSL-KDD dataset, Boruta Shap feature selection yielded 28 features, excluding class labels, that were deemed significant. Similarly, for the UNSW-NB15 dataset, Boruta Shap feature selection identified 19 relevant features.

In our approach, we utilized a deep neural network with a solitary hidden layer containing 50 neurons as our classifier models. We specifically focused on the multiclassification task for both datasets during our evaluation process. To assess the performance of our intrusion detection models, we employed various metrics including accuracy, true positive rate (TPR), false positive rate (FPR) and receiver operating characteristic (ROC). TPR is the percentage of acutal positives correctly predicted wherease FPR is the percentage of negative instances incorrectly classified as positive. ROC is a graphical representation of the trade-off between the TPR and FPR. The ideal scenario is when the curve hugs the top-left corner of the plot, indicating a high TPR and a low FPR. The area under the ROC curve (AUC-ROC) is often used as a single metric to summarize the performance of a classifier.

Following the training and classification stages, we employed SHAP to provide explanations for the predictions made by our model. Specifically, we utilized DeepSHAP to elucidate the importance of features in both the NSL-KDD and UNSW-NB15 datasets.

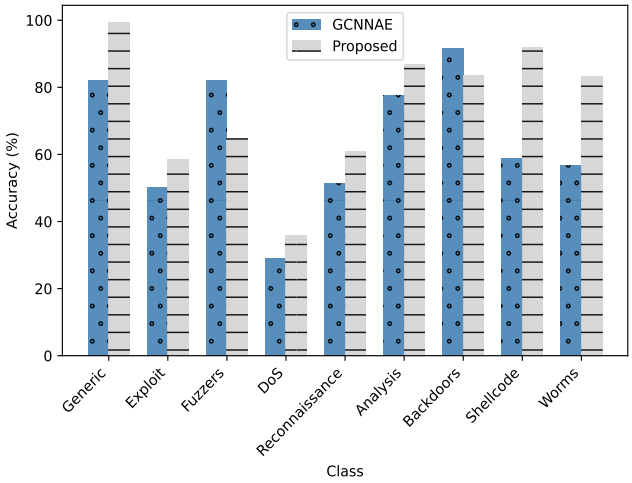

Fig. 4. Multiclass classification results on UNSW-NB15 data set

C. Results

Initially, we applied our classifier to the NSL-KDD datasets without any expansion or feature selection. The resulting ROC curve is depicted in Fig. 2. Subsequently, we employed CTGAN to expand the minority class in both datasets and conducted feature selection using Boruta SHAP. The outcome of this process is illustrated in Fig. 3. In Fig. 2, it can be observed that the area under the curve (AUC) for the R2L and U2R attack classes is relatively low. However, after expanding the dataset, there is a substantial improvement in the AUC for R2L and U2R. These findings demonstrate that our proposed scheme enhances the detection rate for the minority attack class.

In the same manner, we utilized our proposed methodology on the UNSW-NB15 dataset. We conducted a comparison of our results with the study conducted in GCNNAE [17] and obtained the results depicted in Fig. 4. By analyzing the figure,

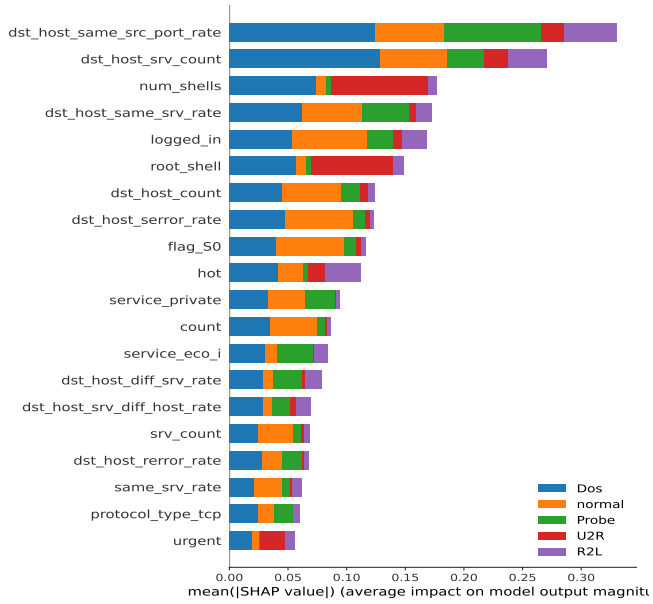

Fig. 5. SHAP feature importance - NSL-KDD

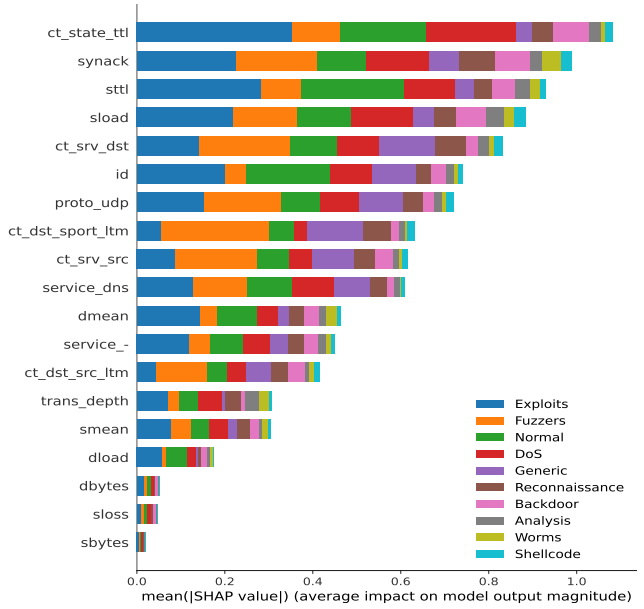

Fig. 6. Shap feature Importance - UNSW-NB15

it becomes evident that our approach achieved higher accuracy for the majority of classes in comparison to GCNNAE [17]. However, we did observe lower accuracy specifically for the classes Fuzzers and Backdoors. We attribute the reason for our higher accuracy to the utilization of CTGAN for dataset expansion and Boruta Shap for feature selection.

1) Feature explanation - NSL-KDD: In our study, we made use of the SHAP summary plot, which is a visual representation demonstrating the overall significance of features in a model. This plot presents the most important features at the top and the least important features at the bottom. In Fig. 5, we displayed a visualization of the top 20 flow features of NSL-KDD dataset that contribute to the prediction of different classes such as DoS, Probing, R2L, and U2R. Observing Fig. 5, it reveals that the feature with the highest weight in relation to the DoS attacks is 'dst_host_srv_count'. This feature represents the number of connections that have the same port number. An attacker can deplete the available resources associated with the port by launching a massive number of connections, resulting in service denial. Thus, this feature is important for detecting DoS attacks confirming the result obtained from SHAP. Another closely ranked feature in terms of weighted SHAP value for DoS attacks is 'dst_host_same_src_port_rate'. This feature measures the rate at which connections from the same source port are established with a particular destination host.

Similarly in Fig. 5, probe attacks show an association with the 'dst_host_same_src_port_rate' in terms of weighted SHAP values. Classic probe attacks often involve sending multiple requests from a single source IP address to the target system, thus confirming the result from SHAP analysis.

In Fig. 5, U2R attacks demonstrate a similar weighted SHAP value for two features: 'num_shells' and 'root_shells'. U2R attacks refer to specific types of attacks where an unauthorized user aims to obtain root-level access to a system. The feature 'num_shells' represents the count of shell prompts or command-line interfaces open on the system. On the other hand, the 'root_shells' feature pertains to the presence of a shell with root-level privileges. Both 'num_shells' and 'root_shells' serve as reliable indicators of privilege escalation attempts and high weighted SHAP values highlight their significance in identifying U2R attacks.

Similarly, the features 'dst_host_same_src_port_rate', and 'dst_host_srv_count' are found to be appropriate for detecting R2L attacks.

2) Feature explanation - UNSW-NB15: As shown in Fig. 6, in relation to exploits, the two most important features are 'ct_state_ttl' and 'sttl'. The feature 'ct_state_ttl' refers to the connection state value of time to live (TTL) which represents the TTL value associated with the connection state of a network packet. The second important feature, 'sttl', stands for source time to live. In the context of exploits, these features are crucial as they help in analyzing the behavior and characteristics of network connections related to potential exploitation attempts.

In the case of fuzzers, the most important features from Fig. 6 are 'ct_dst_sport_ltm', 'ct_srv_dst', and 'ct_srv_src'. The feature 'ct_dst_sport_ltm' refers to the count of the destination source port being repeated in a short time window. This feature helps identify instances where the same source port is repeatedly used for connections to the destination. Fuzzers often exhibit this behavior by sending multiple requests from the same source port in quick succession. 'ct_srv_dst' represents the count of connections made to the same destination service. Fuzzers may target a particular service repeatedly, leading to a higher count for this feature. Similarly, 'ct_srv_src' denotes

the count of connections originating from the same source service.

In the case of DoS attacks, the most important features are identified as 'ct_state_ttl','synack', and 'sload'. The feature 'ct_state_ttl' has been determined to be significant in the context of DoS attacks. Analyzing the TTL values associated with the connection states helps in identifying potential DoS attacks, as abnormal or malicious TTL values can indicate suspicious network behavior. The 'synack' feature refers to the occurrence of SYN-ACK packets in a network connection. Monitoring the frequency and patterns of SYN-ACK packets assists in recognizing potential DoS attacks that involve overwhelming a target system with excessive connection requests. Additionally, the 'sload' represents the server load or load on the target server can be indicative of a DoS attack.

For reconnaissance activities, two important features are 'synack' and 'ct_srv_dst'. Reconnaissance activities involve probing different services on a target system to gather information about potential vulnerabilities or weaknesses. By analyzing the presence of SYN-ACK packets and monitoring the count of connections to the same destination service, it becomes possible to identify reconnaissance activities, thus confirming the result obtained from SHAP. Similar analysis can be done for the backdoor, analysis, worms, and shellcode attacks. Thus, model explainability obtained using SHAP can be employed by security analysts to enhance their comprehension of attacks. Moreover, they can identify flaws in the model's logic, allowing them to enhance its effectiveness.

V. CONCLUSION

In this paper, we introduced an improved and interpretable deep learning approach for intrusion detection in IoT networks. Our proposed method addresses the challenges of data imbalance and interpretability commonly encountered in deep learning-based IDS. To handle the data imbalance issue, we utilized CTGAN for generating synthetic samples for the minority class traffic. Additionally, we employed Boruta Shap for feature extraction, resulting in a reduced number of features and efficient detection. Furthermore, we incorporated SHAP for model explainability. We validated the results obtained from SHAP by conducting an analysis of each attack type in both the NSL-KDD and UNSW-NB15 datasets. Furthermore, we conducted a comparative evaluation of our framework against the previous approach and found that our proposed framework outperforms the previous framework in terms of accurately detecting attacks for the majority of classes.

REFERENCES

[1] F. Meneghello, M. Calore, D. Zucchetto, M. Polese, and A. Zanella, "Iot: Internet of threats? a survey of practical security vulnerabilities in real iot devices," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8182–8201, 2019.

[2] T. Alladi, V. Chamola, B. Sikdar, and K.-K. R. Choo, "Consumer iot: Security vulnerability case studies and solutions," *IEEE Consumer Electronics Magazine*, vol. 9, no. 2, pp. 17–25, 2020.

[3] S. Gyawali, T. Shimizu, H. Lu, M. Clifford, J. Kenney, and Y. Qian, "Local perception and bsm based misbehavior detection in intelligent transportation system," in *2022 IEEE 96th Vehicular Technology Conference (VTC2022-Fall)*, 2022, pp. 1–5.

[4] S. Gyawali and O. Beg, "Cyber attacks detection using machine learning in smart grid systems," in *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2022, pp. 1–2.

[5] J. Huang, J. Lv, Z. Zhou, S. Gyawali, and Y. Qian, "A multi-objective model for misbehavior detection in iov," in *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, 2022, pp. 4395–4400.

[6] N. Jaton, S. Gyawali, and Y. Qian, "Distributed neural network-based ddos detection in vehicular communication systems," in *2023 16th International Conference on Signal Processing and Communication Systems (ICSPCS), Bydgoszcz, Poland*, 2023, pp. 1–5.

[7] S. Gyawali, Y. Qian, and R. Q. Hu, "Machine learning and reputation based misbehavior detection in vehicular communication networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 8, pp. 8871–8885, 2020.

[8] L. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni, "Modeling tabular data using conditional gan," in *Advances in Neural Information Processing Systems*, vol. 32. Curran Associates, Inc., 2019.

[9] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Advances in neural information processing systems*, vol. 30, 2017.

[10] I. A. Khan, N. Moustafa, D. Pi, W. Haider, B. Li, and A. Jolfaei, "An enhanced multi-stage deep learning framework for detecting malicious activities from autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 25 469–25 478, 2022.

[11] N. Moustafa and J. Slay, "Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set)," in *2015 Military Communications and Information Systems Conference (MilCIS)*, 2015, pp. 1–6.

[12] Y. Yan, L. Qi, J. Wang, Y. Lin, and L. Chen, "A network intrusion detection method based on stacked autoencoder and lstm," in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, 2020, pp. 1–6.

[13] G. Muhammad, M. S. Hossain, and S. Garg, "Stacked autoencoder-based intrusion detection system to combat financial fraudulent," *IEEE Internet of Things Journal*, vol. 10, no. 3, pp. 2071–2078, 2023.

[14] M. Tavallaee, E. Bagheri, W. Lu, and A. 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.

[15] Z. A. E. Houda, B. Brik, and L. Khoukhi, ""why should i trust your ids?": An explainable deep learning framework for intrusion detection systems in internet of things networks," *IEEE Open Journal of the Communications Society*, vol. 3, pp. 1164–1176, 2022.

[16] A. Oseni, N. Moustafa, G. Creech, N. Sohrabi, A. Strelzoff, Z. Tari, and I. Linkov, "An explainable deep learning framework for resilient intrusion detection in iot-enabled transportation networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 1, pp. 1000–1014, 2023.

[17] C. Park, J. Lee, Y. Kim, J.-G. Park, H. Kim, and D. Hong, "An enhanced ai-based network intrusion detection system using generative adversarial networks," *IEEE Internet of Things Journal*, vol. 10, no. 3, pp. 2330–2345, 2023.

[18] G. Zhang, X. Wang, R. Li, Y. Song, J. He, and J. Lai, "Network intrusion detection based on conditional wasserstein generative adversarial network and cost-sensitive stacked autoencoder," *IEEE Access*, vol. 8, pp. 190 431–190 447, 2020.

[19] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proceedings of the 34th International Conference on Machine Learning*, vol. 70. PMLR, 06–11 Aug 2017, pp. 214–223.

[20] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved training of wasserstein gans," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc., 2017, p. 5769–5779.

[21] M. B. Kursa and W. R. Rudnicki, "Feature selection with the Boruta package," *Journal of Statistical Software*, vol. 36, no. 11, pp. 1–13, 2010. [Online]. Available: https://doi.org/10.18637/jss.v036.i11

[22] E. Keany, "BorutaShap : A wrapper feature selection method which combines the Boruta feature selection algorithm with Shapley values." Nov. 2020. [Online]. Available: https://doi.org/10.5281/zenodo.4247618

[23] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning important features through propagating activation differences," in *International Conference on Machine Learning*, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:3385018