Proceedings of 14th IOE Graduate Conference
Peer Reviewed
Year: 2023 Month: December Volume: 14
ISSN: 2350-8914 (Online), 2350-8906 (Print)

Explainable NIDS: An Ensemble Approach Using XGBoost, SHAP Explanation and Autoencoders

Anup Regmi [a], Lok Nath Regmi [b], Nanda Bikram Adhikari [c]

[a, b, c] *Department of Electronics and Computer Engineering, Pulchowk Campus, IOE, Tribhuvan University, Nepal*
[a] 078mscsk005.anup@pcampus.edu.np, [b] lnregmi046@pcampus.edu.np, [c] adhikari@ioe.edu.np

Abstract
This research paper introduces an innovative approach to network intrusion detection aimed at identifying both frequently occurring known attack types and unusual traffic patterns transparently and interpretably. The proposed model combines the strengths of supervised and unsupervised learning techniques while leveraging interpretability tools for enhanced insight. In supervised learning, we employ XGBoost as our primary algorithm, augmented by SHAP (SHapley Additive exPlanations) explanations to shed light on the inner workings of our model by transforming input feature vectors into vectors of feature importance scores for each class, thereby facilitating the understanding of prediction rationale. In the next phase, unsupervised learning methods is harnessed by using auto-encoder. The auto-encoder play a crucial role in distinguishing anomalous traffic flows and detecting normal flows and known attacks, enhancing the model's overall versatility and adaptability. We conduct comprehensive testing to evaluate the model's effectiveness using diverse datasets, combining NF-UNSW-NB15-v2, NF-ToN-IoT-v2, NF-BoT-IoT-v2, and NF-CSE-CIC-IDS2018-v2. The results of our experiments demonstrate above 99% of detection rate for anomalous traffic patterns or potential zero-day attacks against both normal and the combination of normal and known attacks. Importantly, our model's overall performance compares favourably with state-of-the-art approaches documented in the cybersecurity literature.

Keywords
NIDS, XAI, SHAP, XGBoost, Autoencoder, Ensemble, Explainable, Interpretable

1. Introduction

In today's digitally connected world, the expansion of networked systems has ushered in an era marked by unparalleled convenience and efficiency. However, this heightened connectivity has also exposed these systems to cybersecurity threats, ranging from sophisticated malware to targeted intrusion attempts. In response to these challenges, Network Intrusion Detection Systems (NIDS) have emerged as a vital line of defence for identifying and mitigating malicious activities within network traffic.

While NIDS have demonstrated their effectiveness in pinpointing anomalies and potential threats, they have also presented a fundamental challenge: their inherent black-box nature. This opacity has left cybersecurity practitioners grappling with a critical issue—comprehending the rationale behind NIDS decisions. This lack of transparency hampers the ability to trust these systems, fine-tune their performance, and ultimately enhance their detection capabilities. As a result, there is a pressing demand for NIDS solutions that provide robust security and clear and comprehensible explanations for their decision-making processes. These limitations step into the domain of explainable NIDS [1]. This research paper introduces an innovative approach to building Explainable NIDS, capitalizing on a powerful ensemble strategy that harnesses the strengths of XGBoost, SHAP explanations, and Autoencoders[2]. The overarching goal is twofold: to elevate the detection accuracy of NIDS through the synergy of machine-learning models and to demystify the decision-making process, making it interpretable for cybersecurity analysts.

The ensemble methodology outlined in this paper leverages the versatility of XGBoost[3], a gradient-boosting framework celebrated for its exceptional predictive capabilities. We aim to illuminate the underlying logic governing the ensemble's predictions by integrating SHAP [4] explanations. SHAP explainers provide valuable insights into feature importance and contribution, enhancing transparency. Additionally, Autoencoders, a neural network variant, come into play to capture intricate patterns within network traffic data. This further fortifies the ensemble's ability to accurately discern and flag malicious behaviour.

In the forthcoming sections, we delve into the technical intricacies of our Explainable NIDS approach, which includes an in-depth exploration of the architecture, data preprocessing steps, and training strategies. Moreover, we present a comprehensive set of experimental results, underscoring the advantages of our ensemble model. In essence, this paper seeks to elevate the effectiveness and efficiency of threat detection, facilitate more informed decision-making, and foster collaborative human-AI cybersecurity operations by infusing NIDS with the power of explainability.

2. Related Works

Several noteworthy research efforts have contributed to advancing NIDS and the quest for accurate and explainable multi-class classification of network traffic data. Giuseppina et al. [5] introduced a novel neural model attention-based method, focusing on achieving precise and interpretable multi-class classification. Maonan et al. [6] proposed a

comprehensive framework that extends the scope of intrusion detection systems by incorporating SHAP explanations. This framework combines both local and global explanations to enhance the interpretability of IDS, shedding light on the decision-making process. Zakaria et al. [7] took a multifaceted approach, designing a DL and XAI-based system. They leveraged three distinct explanation methods, namely LIME, SHAP, and RuleFit, to provide local and global explanations for the outputs of a DNN model.

Pieter et al. [8] presented a two-stage pipeline for binary classification tasks involving normal and suspicious network traffic. Their approach initially utilises XGBoost for the first-phase classification, followed by autoencoders. SHAP explanations derived from the XGBoost model are then fed into the autoencoder for anomaly detection. This innovative methodology demonstrated remarkable performance improvements, excelling in accuracy, recall, and precision scores compared to alternative models.

Lopez-Martin et al. [9] introduced a classification model employing a conditional variational autoencoder that detects and categorises different label types within network traffic data. Mirsky et al. [10] adopted an ensemble approach to develop an ML-NIDS named Kitsune. Their proposed model transforms network packet features into an ensemble of autoencoders, with each autoencoder responsible for reconstructing packet features and computing the Root Mean Square Error (RMSE). This process enables the classification of network traffic based on predefined thresholds, enhancing the system's overall effectiveness in intrusion detection. These pioneering studies represent significant contributions to network intrusion detection, showcasing innovative techniques and methodologies to improve the accuracy, interpretability, and overall performance of NIDS.

Upon delving into the existing literatures, it becomes clear that models constructed through the orchestration of Supervised Network Intrusion Detection Systems, eXplainable Artificial Intelligence (XAI), and Anomaly-Based NIDS exhibit an enviable array of superior performance metrics. These encompass essential measures like accuracy, precision, and recall scores. These models notably showcase a remarkable aptitude for zero-day attack detection—an essential capability in modern cybersecurity. It is worth highlighting, however, that these models currently find themselves constrained within the confines of binary classification, thereby limiting their capacity to offer detailed insights into specific attack types. We have embraced a refined approach to overcome these inherent constraints, using XGBoost as the primary classifier to identify the normal and known attack types. Further, autoencoder is trained to distinguish the anomalous traffic flows. This harmonious synergy empowers our model with the unique capability to recognize well-known attack types and discern the subtle intricacies of anomalous network traffic flows.

3. Methodology

3.1 Proposed Model

Our model is meticulously crafted to adeptly classify benign flows and other five specific attack classes :DDoS , DoS, XSS, scanning, and Reconnaissance as known attack types. Remarkably, our model extends its adaptability to encompass the ever-evolving threat landscape by categorizing all other attack types within the dataset as new or zero-day threats—a testament to its agility in addressing emerging risks. For comparision of performance of autoencoder in different training set, it is fed with feature vectors as training set in one environment and in other case we used the SHAP explanation to train the autoencoder. The block diagram of these test environment is as shown in figure 1 and figure 2.

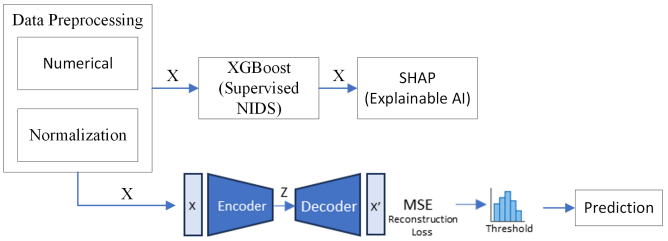

Figure 1: Block Diagram of Model to use Feature Vector to train the Autoencoder

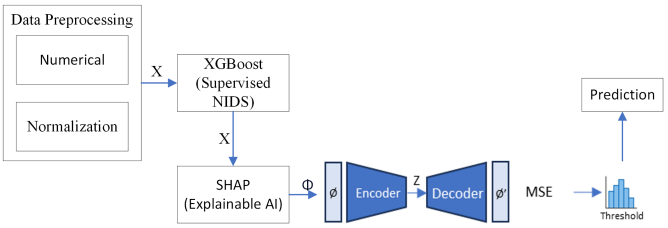

Figure 2: Block Diagram of Model to use SHAP values to train the Autoencoder

Proposed model is subjected to rigorous testing in the subsequent validation phase using a dedicated dataset split. Furthermore, introducing new attack types served as a litmus test, evaluating its capacity to detect known and novel attack patterns effectively. This comprehensive evaluation underscores the model's robustness and establishes it as a formidable asset in contemporary network security—capable of addressing the ever-evolving threat landscape with confidence and competence.

3.2 Implementation

In the data collection phase, to assess and compare various attack scenarios conducted across diverse testbed networks, we utilize the combination of various datasets: NF-UNSW-NB15-v2 [11], NF-ToN-IoT-v2[11], NF-BoT-IoT-v2[11], and NF-CSE-CIC-IDS2018-v2[11]. This datasets comprises a rich collection of 43 NetFlow-v9 features meticulously extracted from their respective pcap files. Approximately 33% of the data in this dataset represents benign network flows, while the remainder encompasses a diverse spectrum of attack categories. This comprehensive amalgamation provides a robust and multifaceted foundation for evaluating and analysing various attack scenarios across different network environments. In the preprocessing stage we removed the duplicate rows, rows with infinte values and

not-a-number values and defined frequently occurring known attacks as those that have been observed more than a million times. These well-documented attack types are categorized and classified as known attack class by our model. However, it's important to note that the remaining attack types, which occur less frequently, are treated as suspicious traffic or potential zero-day attacks. These are the types of attacks that have not been extensively documented or observed, making them a critical focus for anomaly detection. Table 1 shows the distribution of attack types in the dataset and their respective frequencies. This information serves as a valuable reference for understanding the dataset's composition and the focus of our analysis.

Table 1: Categorywise distribution of attack types in dataset

SN	Attack Label	No. of traffic flow
1	Benign	20744072
2	DDoS	17302920
3	DoS	14787587
4	Scanning	3002169
5	XSS	2449955
6	Reconnaissance	2374188
7	Others	2011774

In the subsequent stages of our approach, we undertake a series of data preprocessing steps to prepare our dataset for the classification phase. These steps are essential to ensure the fairness and uniformity of our data representation. First, we address potential bias by dropping source and destination IP addresses from our dataset. This step helps remove source-specific or destination-specific patterns, ensuring a more balanced and unbiased dataset. Next, we employ label encoding for attack types to ensure uniformity in our data representation.

Additionally, we employ a crucial normalization step using the min-max scaler. Through this process, we scale each feature to a standardized range of (0, 1), ensuring consistent and fair data treatment. Our focus for the initial classification phase narrows down to normal flows and frequently occurring attacks. We separate the remaining less frequent attack types from the training and testing datasets to facilitate this. The dataset is then divided into an 80-20 percentage of train and test split for this preliminary classification stage.

In this stage, we harness the power of the XGBoost classifier. We employ the multiclasslogloss metric during training to evaluate and refine the model's performance. Notably, we set the use label encoder parameter to false. This choice is deliberate, as we have already encoded the attack types in prior stages, ensuring the model can learn effectively and accurately refine its predictions.

The XGBoost classifier becomes the focal point of our analysis, and we delve into its inner workings through the SHAP tree explainer. This enlightening explainer is instrumental in shedding light on the inner workings of the classifier by providing explanations in the form of SHAP values. Each SHAP value corresponds to a traffic flow and represents a vector with both magnitude and direction. These values elucidate the role of each feature in the model's decision-making process, offering a clear picture of how each feature contributes to the predictions made by the classifier. In essence, the SHAP values generate a feature importance vector, which allows us to discern the relative significance of different features in shaping the predictions of our model. The feature vector $x = [x_1, x_2, \ldots, x_{41}]$ is converted into the SHAP values $\phi = [[\phi_1, \phi_2, \ldots, \phi_{41}]$ is the feature importance of x_1 to identify 'x' as class '1', and ϕ_{41} is the feature importance of x_{41} to classify 'x' as class '1' or known attack. It gives us a refined understanding of the underlying factors driving the classifier's decisions for various attack classes. By harnessing the power of SHAP explanations, we equip ourselves with the knowledge required to interpret the model's output effectively. This interpretability is a critical component of our approach, enabling us to make accurate predictions and understand why those predictions are made, paving the way for transparent and trustworthy network intrusion detection.

Equipped with this enriched vector containing feature importance scores for the input vector, we train the respective autoencoder as defined earlier with both these vectors . These autoencoder is structured with an encoder-decoder framework, featuring layers with 41-20-10-5 neurons and ReLU activation functions for the encoder layers. In contrast, the decoder layers consist of 5-10-20-41 neurons with sigmoid activation function in hidden layers and hyperbolic tangent (tanh) activation function in output layer. To further enhance the autoencoders' performance, we utilize the 'Adam' optimizer as the default choice and meticulously quantify the reconstruction loss using the Mean Squared Error (MSE) metric as given by following equation:

$$\text{MSE} = \sum_{i=1}^{N} (x_i - x_i')^2$$

This holistic approach ensures our model excels in classifying known attack types and demonstrates resilience when confronted with emerging threats. Individual autoencoder model is trained for benign flows and frequently occurring attack types over 50 epochs, employing mini-batching with a size of 128. For anomaly detection, the reconstruction loss on the training data is evaluated, and a threshold is set as the mean of the training loss plus its five times it's standard deviation. This threshold serves as a decisive boundary: incoming traffic flows with a reconstruction loss below this limit are classified as frequently occuring attack types. In contrast, those exceeding the threshold are flagged as anomalous flows.

4. Result and Analysis

After training the XGBoost model with the training set and evaluating its performance on the test set, we obtained the classification report presented in Table 2. Additionally, the confusion matrix is illustrated in Figure 3. Table 2 shows that the model performs exceptionally well regarding accuracy, precision, recall, and F1-score across normal flow and known attack types. This exceptional performance is necessary for the subsequent modules, where accurate classification is critical. It is worth noting that the high precision values indicate a low false alarm rate, signifying that when the model predicts an intrusion, it is highly likely to be a genuine threat. Conversely, the high recall values indicate a high detection

rate, emphasizing the model's effectiveness in correctly identifying intrusions. These evaluation metrics are essential in intrusion detection systems, where minimizing false alarms and maximizing threat detection are primary objectives.

Table 2: Classification Report of XGBoost Model

Class	Precision	Recall	F1-Score	Support
0	0.99	0.99	0.99	41564
1	0.99	0.99	0.99	79758
Accuracy			0.99	121322
macro avg	0.99	0.99	0.99	121322
weighted avg	0.99	0.99	0.99	121322

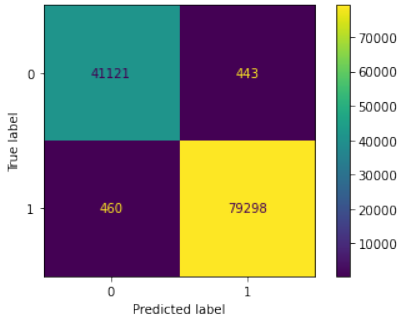

Figure 3: Confusion Matrix of XGBoost Model

Figure 4 provides a global explanation of XGBoost Classifier used in previous step. This explanation is generated by the SHAP tree explainer applied to the trained XGBoost model. The figure shows the impact of each feature on the decision to be positive or as known attack. The features impacting the most are placed at the top while less important one are lower down. Horizontal line in the middle represent the baseline prediction, which is the expected value of prediction of the benign flow, the points above this line contribute positively and below this line contribute negatively. In the figure DURATION_IN is the most impacful feature and CLIENT_TCP_FLAGS is the least impactpul featue. Furthermore, the figure showcases the top 20 features, their respective magnitudes, and directions of influence on the classification. These values represent the SHAP values or feature importance scores, providing valuable insights into which features play the most crucial roles in determining the class of traffic flow.

In the anomaly-based NIDS phase, we employed autoencoder, trained to detect seen and unseen network traffic patterns. The autoencoder was trained using the input feature vector and feature importance scores provided by the SHAP explainer derived from our XGBoost model's training data and with input feaure vector that denotes the benign flow.

Autoencoder learned to capture the unique characteristics and anomalies associated with the benign flows during training. We established a threshold based on the reconstruction loss of each autoencoder to determine whether incoming traffic was anomalous or normal. For testing, we evaluated each autoencoder's performance using two types of data:

Normal Flow: We used the input feature and feature importance scores of the normal traffic flows to train the

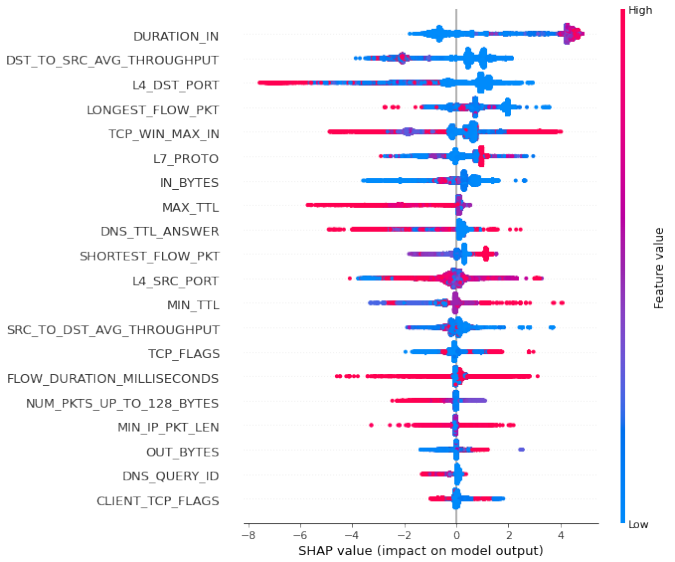

Figure 4: Global explanation for prediction made by XGBoost Classifier

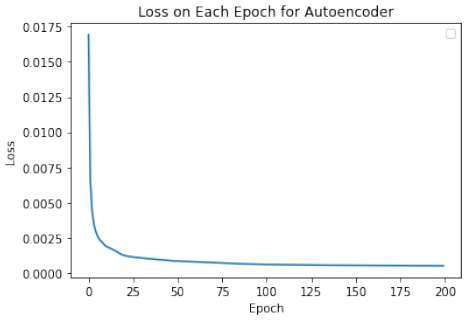

Figure 5: Loss on each epoch of autoencoder trained with input feature vector

respective autoencoder. This allowed us to assess the autoencoder's ability to detect normal flows effectively.

Unseen Attack Types: To evaluate the adaptability of our autoencoder, we tested them with the input feature and feature importance scores of previously unseen attack types. This allowed us to assess the autoencoder's ability to detect the previously unseen attack or potential zero-day threats.

Training loss in each epochs for each autoencoders is as shown in figure 5 and 6. The performance of each autoencoders is depicted in figures 7 and 8, providing insights into their capabilities to detect normal traffic flow, and avoid unseen attack types. The detection or avoidance is shown by the threshold, red dotted line in figures. These results are instrumental in assessing the overall effectiveness of our network intrusion detection system.

To quantitatively assess the performance of our autoencoders, we conducted two distinct test scenarios as mentioned below:

For distinguishing unseen new attacks from the normal traffic, method as shown in figure 1 is used, where autoencoder one is trained with the feature vector of traffic flows belonging to the normal traffic. This is tested with the set of new unseen attack types as mentioned as others in table 1. The prediction of the model against its true label are as shown in figure 9. The

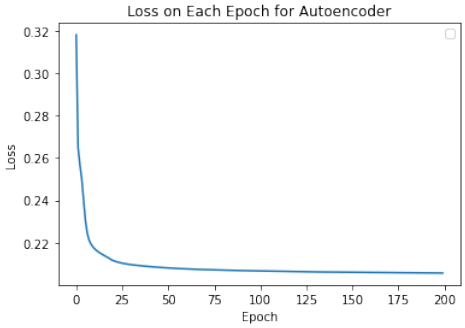

Figure 6: Loss on each epoch of autoencoder trained with SHAP values

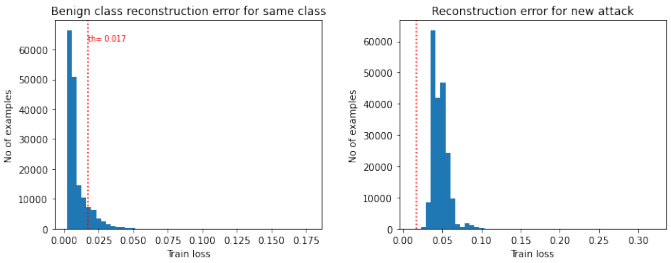

Figure 7: Performance of autoencoder trained with the feature vector of normal traffic flow

detection rate for this model is observed as 1, and the false alarm rate is observed as 0.138.

In another scenario, the autoencoder is used to distinguish new attacks from the previosuly seen attack for this the method as shown in figure 2 is used where autoencoder is trained with the SHAP values obtained from XGBoost explainer. The model is tested with the set of new unseen attack types as mentioned as others in table 1. The prediction of the model against its true label are as shown in figure 10. The detection rate for this model is observed as 0.991, and the false alarm rate is observed as 0.138.

While the detection rates for each of the autoencoders demonstrate exceptional performance, there is room for improvement in terms of enhancing the false alarm rates. It's worth noting that any traffic flows that escape detection by the autoencoders are flagged as suspicious. These instances of suspicious traffic flows are crucial and require manual investigation by security analysts. This investigative step is necessary to identify and assess the potential threat posed by these flows, as they may represent emerging attack patterns or zero-day vulnerabilities. In this way, the autoencoders serve as

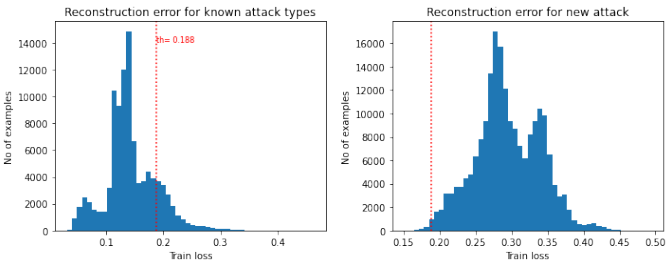

Figure 8: Performance of autoencoder trained with the SHAP values of normal traffic flow

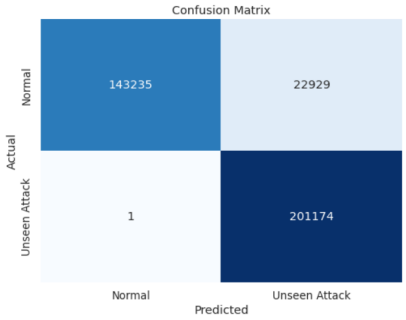

Figure 9: Prediction against the true label for autoencoder trained with input feature vector

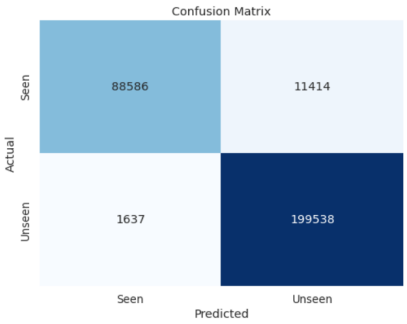

Figure 10: Prediction against the true label for autoencoder trained with SHAP values

a valuable first line of defense, effectively filtering out benign traffic, and raising red flags when there is a strong indication of anomalous behavior.

5. Conclusion and Future Work

In conclusion, our research has successfully implemented an ensemble approach to building an explainable Network Intrusion Detection System (NIDS). This approach leverages hybrid learning methods, combining the power of XGBoost, autoencoder, and an explainable layer to enhance the detection of known attack types.

Our experimental evaluation, conducted across a diverse range of datasets, including NF-UNSW-NB15-v2, NF-ToN-IoT-v2, NF-BoT-IoT-v2, and NF-CSE-CIC-IDS2018-v2, underscores the robust operability of our model in various network scenarios. Importantly, our model exhibits the capacity to identify known attack while also effectively flagging previously unseen attack patterns as anomalous traffic flows.

The performance of our overall model, as well as the individual components within it, is highly competitive when compared to state-of-the-art cybersecurity research. In several instances, our approach has even surpassed the performance of existing methodologies, reinforcing the effectiveness and relevance of our ensemble-based, explainable NIDS in the ever-evolving landscape of network security. This research opens new avenues for building more robust and interpretable network intrusion detection systems, contributing to the ongoing efforts to protect critical networks from cyber threats.

In the future, there are several exciting avenues for enhancing

this research. Firstly, we can increase the known attack domain by classifying them to the respective class of attack, which will be one step closer to interpretablity of the model. Next step is to design the single point for the prediction and explanation which involve consolidating the predictions of different modules inside the model. Moreover, there is potential for further advancements in the explanation domain, allowing us to interpret individual explanations for traffic flows to detect anomalies more effectively. This opens the door to qualitative evaluations of our model's effectiveness in cybersecurity, ultimately improving our understanding and response to network threats.

Acknowledgments

The authors express sincere gratitude to the Department of Electronics and Computer Engineering, Pulchowk Campus for their unwavering support and resources throughout the course of this research. Their commitment to fostering a vibrant academic environment and providing us with access to cutting-edge facilities has been instrumental for the completion of this research.

References

[1] Nicola Capuano, Giuseppe Fenza, Vincenzo Loia, and Claudio Stanzione. Explainable artificial intelligence in cybersecurity: A survey. *IEEE Access*, 10:93575–93600, 2022.

[2] Youngrok Song, Sangwon Hyun, and Yun-Gyung Cheong. Analysis of autoencoders for network intrusion detection. *Sensors*, 21(13), 2021.

[3] Tianqi Chen and Carlos Guestrin. XGBoost. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, aug 2016.

[4] Scott Lundberg and Su-In Lee. A unified approach to interpreting model predictions, 2017.

[5] Giuseppina Andresini, Annalisa Appice, Francesco Paolo Caforio, Donato Malerba, and Gennaro Vessio. Roulette: A neural attention multi-output model for explainable network intrusion detection. *Expert Systems with Applications*, 201:117144, 2022.

[6] Maonan Wang, Kangfeng Zheng, Yanqing Yang, and Xiujuan Wang. An explainable machine learning framework for intrusion detection systems. *IEEE Access*, 8:73127–73141, 2020.

[7] Zakaria Abou El Houda, Bouziane Brik, and Lyes 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*, 3:1164–1176, 2022.

[8] Pieter Barnard, Nicola Marchetti, and Luiz A. DaSilva. Robust network intrusion detection through explainable artificial intelligence (xai). *IEEE Networking Letters*, 4(3):167–171, 2022.

[9] Manuel Lopez-Martin, Belen Carro, Antonio Sanchez-Esguevillas, and Jaime Lloret. Conditional variational autoencoder for prediction and feature recovery applied to intrusion detection in iot. *Sensors*, 17(9), 2017.

[10] Yisroel Mirsky, Tomer Doitshman, Yuval Elovici, and Asaf Shabtai. Kitsune: An ensemble of autoencoders for online network intrusion detection, 2018.

[11] Mohanad Sarhan, Siamak Layeghy, and Marius Portmann. Towards a standard feature set for network intrusion detection system datasets. *Mobile Networks and Applications*, 27(1):357–370, nov 2021.