Machine Learning based Intrusion Detection System for IoT Applications using Explainable AI

Muhammad Asim Mukhtar Bhatti
Military College of Signals, National University of Sciences and Technology Islamabad Pakistan
asimmukhtar1146@gmail.com

Muhammad Awais
Military College of Signals, National University of Sciences and Technology Islamabad Pakistan
muhammad.awais@hotmail.com

Aamna Iqtidar*
College of Electrical and Mechanical Engineering, National University of Sciences and Technology Islamabad Pakistan
aamna.iqtidar@gmail.com

ABSTRACT

This research focuses on studying the classification performance of a Machine Learning-based Intrusion Detection System (IDS) using the UNSW-NB15 dataset. The effectiveness of three classifiers - Decision Tree, Multilayer Perceptron (MLP), and XGBoost - was analyzed to determine their accuracy in identifying attacks and normal network traffic. The experimental results revealed that Decision Tree achieved an accuracy of 96.5%, MLP achieved an accuracy of 89.83%, and XGBoost achieved an accuracy of 89.9%. Additionally, the Explanability of the machine learning models was analyzed, highlighting the differences in interpretability among the classifiers. It was observed that Decision Tree provided better Explanability, but lower accuracy compared to MLP and XGBoost. Overall, this research contributes to our comprehension of the performance and Explanability of three different machine learning classifiers for intrusion detection. The findings can provide valuable insights for choosing suitable classifiers that align with the specific priorities and requirements of the IDS system.

CCS CONCEPTS

• :; • **Computing methodologies** → Machine learning algorithms; • **Security and privacy** → Intrusion detection systems;

KEYWORDS

Intrusion detection system (IDS), Artificial Intelligence (AI), Machine learning (ML), Decision Tree, Multilayer Perceptron (MLP), XGBoost classifier, Internet of things

ACM Reference Format:
Muhammad Asim Mukhtar Bhatti, Muhammad Awais, and Aamna Iqtidar. 2023. Machine Learning based Intrusion Detection System for IoT Applications using Explainable AI. In *2023 Asia Conference on Artificial Intelligence, Machine Learning and Robotics (AIMLR 2023), September 15–17, 2023, Bangkok, Thailand.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3625343.3625356

*Corresponding Author

AIMLR 2023, September 15–17, 2023, Bangkok, Thailand

ACM ISBN 979-8-4007-0831-2/23/09…$15.00
https://doi.org/10.1145/3625343.3625356

1 INTRODUCTION

An Intrusion Detection System (IDS) plays a crucial role by constantly monitoring network traffic, system logs, and various data sources to identify and address any suspicious activities for Internet of Things (IoT) devices. The AI-powered explainable IDS assists cybersecurity analysts in comprehending the rationale behind identifying specific events or activities as potential threats. This capability offers valuable insights into the features, indicators, or behaviours that influenced the system's decision, enabling analysts to validate the accuracy of the system's findings, verify any false positives or negatives, and make informed decisions accordingly and also helps in making adjustments to detection rules. Furthermore, Explainability significantly enhances the effectiveness of incident response and forensic investigations and

Help analysts int tracing the attack path, comprehending the attacker's tactics, techniques, and motives, and developing countermeasures to prevent future incidents.

In general, the IDS can be categorized into two main types: Host-based IDS (HIDS) and Network-based IDS (NIDS) [1]. NIDS, focuses on analyzing and monitoring network traffic to identify malicious activities. Whereas, HIDS detects and prevents malevolent actions that occur within the operating system files [2]. The development approaches of IDS are divided into traditional [3] and machine/deep learning [4]. When it comes to traditional tactics, the utilization of deep/machine learning techniques in intrusion detection leads to higher detection rates, as pointed out by Chen et al. [5].

Cybersecurity researchers rely on IDS as a crucial tool to make IoT devices and IoT networks secure. Only a small amount of study and implementation has gone into establishing IDS using machine learning methods and other statistical feature learning algorithms. Nour Mustafa performed research [6] resulting in a ML based IDS solution for IoT network traffic protection based on specified statistical flow properties. An AdaBoost ensemble learning technique was applied to the three ML based algorithms, namely artificial neural network, Naive Bayes (NB) and Decision Tree. Using data from simulated IoT sensors, the created models' ability to identify harmful events was evaluated on the NIMS botnet datasets and UNSW-NB15. Detecting both normal and harmful behavior, the proposed ensemble methodology offers a larger detection rate, also lower false positive rate than three others prevalent IoT cybersecurity approaches. As noted by Kelton da Costa's review, many further ML-based IDS have been explored and developed [7]. Regarding Intrusion Detection and the IoT, cybersecurity experts use a broad range of machine learning techniques. Approximately 95 papers

were reviewed, covering various IoT machine learning and security concerns.

Reviewing the total research effort into the Explainability of ML approaches may be done with the help of a comprehensive paper [8] that summarizes the important ideas, taxonomies, possibilities, and difficulties of responsible AI. Because of the growing concern about the inexplicability of modern ML methods, such as concatenative ensembles and Deep Neural Networks, XAI has seen significant growth in recent years. They used the NSL-KDD dataset to show how current XAI algorithms derived from SHAP, LIME, Contrastive Explanations Method (CEM), Protidic, and Boolean Decision Rules through Column Generation (BRCG) may successfully boost model transparency.

In [9], Moustafa et al. compared the features of KDD-99 to the characteristics of UNSW-NB15 regarding their ability to identify attacks. NB15's to prioritize features from both datasets, the method used for selection of feature was Association Rule Mining (ARM). For detection of accuracy, clustering using Expectation Maximization (EM) and Naive Bayes (NB) modelling were used. The findings demonstrate that the attack detection efficacy of the features of UNSW-NB15 is higher than the original features of KDD-99. With the original KDD-99 features, the NB and EM models obtained 62.02% and 52.54% accuracy, respectively; using the UNSW-NB15 features, they achieved 78.06% and 58.88% accuracy. To properly compare the UNSW-NB15 and KDD-99 feature sets, in the dataset of UNSW-NB15, the authors should have replicated the features of KDD-99. The heterogeneity of current NIDS datasets was addressed by Sarhan et al. [10], who noted the difficulties in generalizing results. This implies that the proposed ML-based NIDS evaluation methods are generally not reliable. The ability to compare the performance of different dataset of ML model is hindered by the absence of a universally applicable feature set. The solution proposed in this study is creating and publishing four datasets, each of them shares a common set of 12 features based on NetFlow data: NF-CSE-CIC-IDS2018, NF-ToN-IoT, NF-BoT-IoT and NF-UNSW-NB15. As NetFlow characteristics are already included in packet headers, they may be extracted from network traffic much more quickly than sophisticated features that need deep packet inspection. Detection of assaults in the datasets by the ML models is improved by the additional extracted features, which comprise a sufficient number of security events. Generated and annotated datasets are made available for study under the names NF-CSE-CIC-IDS2018-v2, NF-ToN-IoT-v2, NF-BoT-IoT-v2 and NFUNSW-NB15-v2. For future NIDS dataset, its shared collection of features is offered as a benchmark. NetFlow, according to the authors, is a great candidate for a globally standardized feature set due to its broad acceptance, practical application, and scalability. For the utility evaluation of the suggested feature set, the researchers compare the detection accuracy of the newly proposed feature set to both the original feature set of the datasets and the smaller NetFlow datasets developed in a previous study [11], all with the use of an Extra Tree classifier. The results indicate that the proposed NetFlow feature set exhibits significantly superior performance in terms of attack detection accuracy when compared to the other feature sets, using a single ML classifier. Wang et al. [12] set out to do just that—shine some light on the ML-based NIDS's inner workings. This paper represents the pioneering use of the SHAP method to clarify the underlying decision-making and architectural principles of an intrusion detection using NIDS. A ReLU activation function with Deep Feed Forward (DFF) was employed in the research. The research used Deep Feed Forward (DFF) with a ReLU activation function. The classifiers achieved F1 values of 0.792 and 0.807 in and multi-class and binary experiments, respectively. The performance of model has low detection, and the NSLKDD dataset used in the experiment is too old to replicate modern network attacks reliably [13]. The article utilized one hundred "Neptune" assaults as illustrations during the first local explanation phase. The predictions made by the different classifiers relied on a broad range of factors. Finally, the paper enumerates the top 20 critical aspects of each type of attack that was employed in the explanation and investigates them in detail through connected research. According to the research [14], a DFF with an explainable Artificial Intelligence (XAI) architecture may provide more transparency. The DFF model uses three ReLU-executing hidden layers and an output softmax layer. Several XAI approaches, such as Boolean Decision Rules, SHAP, LIME through the Contrastive Explanation and Column Generation (BRCG) Method. The dataset of NSL-KDD was used to verify the authors' claims, despite criticism that KDD-based datasets don't simulate realistic, complex network attacks [13]. The outcomes obtained from SHAP analysis demonstrate that the same value of the SRV rate feature enhances the probability of an attack prediction, whereas a higher value of the feature host error rate is associated with a higher likelihood of a benign prediction. Furthermore, by applying the BRCG method to the dataset, the authors were able to extract model rules with an accuracy of 80%. A local explanation for certain datasets using a method LIME, allowing researchers to ascertain what circumstances adds to generating a non-threatening prediction or an attack. Lastly, the CEM presented how a little change in a single sample of data might affect the prediction.

2 MATERIALS AND METHODS

2.1 Overview

The training dataset UNSW-NB15 will undergo data processing techniques such as cleaning, normalization, and transformation. This processed dataset will then be utilized to train three supervised ML based binary classifiers: XGBoost, Multi-layer Perceptron Neural Network, and Decision Trees. The objective is to classify instances as either Normal (0) or Attack (1). Subsequently, the trained models will be tested using the processed UNSW-NB15 testing dataset, and their performance will be evaluated using scores for accuracy. It is important to note that the described procedure will not involve tuning the classifier or model hyperparameters. The Multi-layer Perceptron Classifier and Decision Trees Classifier from the Scikit-Learn implementation will be utilized, while for the XGBoost Classifier the XGBoost library will be employed. Figure 1 shows the proposed architecture of the intrusion detection system using machine learning.

Once the classifiers are tested and trained, the next step is to generate visual explanations of the classification/ prediction, plots for feature importance and interpretable diagrams. These visualizations will be based on the trained classifiers and aim to detect network traffic behaviour in the testing set. The investigation will

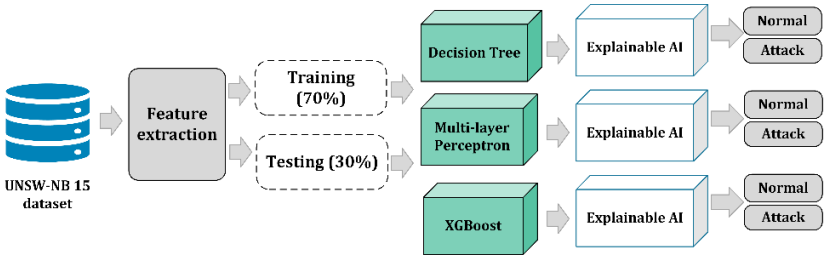

Figure 1: Block diagram of the proposed architecture for machine learning based intrusion detection system

involve exploring Python packages to modify the ML classifiers and enhance their Explainability.

2.2 Dataset: UNSW-NB15

UNSW-NB15 is a dataset for analyzing IoT-based network traffic for normal activities and malicious attack behaviours from botnets (by classifying different type of attacks including Worms, Shellcode, Reconnaissance, Generic, Exploits, DoS, Backdoors, Analysis and Fuzzers). The UNSW-NB 15 dataset consists of raw network packets that were generated using the IXIA PerfectStorm tool in the Cyber Range Lab of the Australian Centre for Cyber Security (ACCS). This tool facilitated the creation of a hybrid dataset that includes both real modern normal activities and synthetic contemporary attack behaviours on IoT-based networks. To capture the raw traffic, the Tcpdump tool was employed, resulting in a collection of 100 GB of data stored in Pcap files.

The UNSW-NB15 is pre-partitioned by its creators into being configured into a testing set for model performance and training set for model training, namely, UNSW_NB15_testing-set.csv and UNSW_NB15_training-set.csv respectively. The testing set comprises 82,332 records and the training set consists of 175,341 records. Each record in both sets includes a target response indicating the traffic behaviour, distinguishing between normal behaviour and an attack. The dataset consists of 49 numeric form of features. The features and their descriptions are listed in the UNSW-NB15_features.csv file. In the experimental procedure, the target feature will be a binary classification distinguishing between Normal and Attack behaviour. It is evident that the dataset exhibits satisfactory balance for the binary response variable representing activity behavior.

2.3 Feature extraction

A raw network traffic of 100 GB was captured by using tcpdump tool (tcpdump tool, 2014). Features were extracted using Bro-IDS (BroIDS Tool, 2014) and Argus (Argus tool, 2014), tools and 12 models. For feature extraction with the class label, these techniques were configured in a parallel processing. The dataset consists of 49 numeric form of features. In-depth description of UNSW-NB15 dataset is as follows. The features are classified into five categories: flow features, basic features, content features, time features, additional generated features.

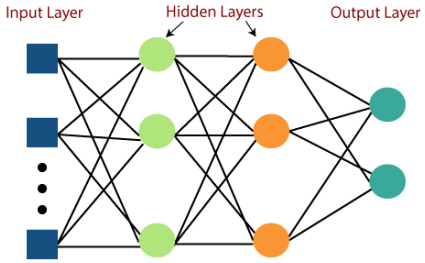

Figure 2: Architecture of the multilayer perceptron (MLP)

2.4 Machine learning based classification

2.4.1 Decision Tree. A decision tree classifier is an algorithm for classification based on supervised machine learning. It constructs a predictive model in the shape of a tree-like structure. In this structure, all internal nodes represent decisions made based on features, branches correspond to the outcomes of these decisions, and leaf nodes represent predictions or class labels.

2.4.2 Multi-layer Perceptron (MLP). A Multi-Layer Perceptron (MLP) is a neural network model that follows a feedforward architecture, comprising multiple layers of interconnected nodes known as "neurons" organized in a sequential manner. It includes an input layer, one or more hidden layers, and an output layer. Each layer contains neurons that apply non-linear activation functions to the inputs. Figure 2 illustrates the architecture of the MLP.

The scikit-learn library offers the MLPClassifier, which is a versatile implementation of the MLP model primarily designed for classification tasks. It provides support for various activation functions, including rectified linear unit (ReLU), hyperbolic tangent (tanh) and logistic (sigmoid) among others. Additionally, it offers different optimization algorithms, such as Adam and stochastic gradient descent (SGD).

2.4.3 Extreme Gradient Boosting (XGBoost). XGBoost is an optimized and powerful gradient boosting framework that is widely used for machine learning tasks, especially in structured data and tabular datasets. It stands for "Extreme Gradient Boosting." XGBoost is known for its speed, scalability, and performance in various data science competitions and real-world applications.

XGBoost is based on the gradient boosting framework, which leverages the combination of multiple weak predictive models, typically decision trees, to generate a robust predictive model. It builds

models in an iterative manner, where each new model focuses on correcting the mistakes made by the previous models.

XGBoost employs tree pruning techniques to control the complexity of individual decision trees. Pruning helps avoid overfitting and ensures that each tree contributes optimally to the overall ensemble model.

2.5 Explainable Artificial Intelligence

Explainable AI (XAI), also known as Explainable Machine Learning (XML) or Interpretable AI, refers to the field of AI where the decisions or predictions made by AI systems can be understood by humans. This stands in contrast to the concept of a black box in machine learning, where even the designers of the AI system are unable to explain the rationale behind specific decisions. In this research, we used three algorithms to add explainability to the machine learning classification models. The details of algorithms for explanability are given below.

ELI5 – ELI5 is a visualization library that may be used to help explain the predictions of ML models and to help debug such models. ELI5 is a Python library for understanding and troubleshooting classification models used in ML.

LIME - The Local Interpretable Model-Agnostic (LIME) Algorithm produces explanations regarding features' contributions in creating a prediction on a single sample, and it accepts as input any machine learning model. It creates an explanation supposing that the model is black box, which indicates that it does not understand how the model works.

SHAP (SHapley Additive exPlanations) - The SHAP method uses game theory to rationalize the results of any ML algorithm. SHAP to learn how various characteristics affect the final model result. SHAP begins with a prediction based on past knowledge, then iteratively tests data characteristics to learn how their addition affects the starting point.

3 RESULTS AND DISCUSSIONS

The original data was divided into a training set of 70% and a testing set of 30%. However, UNSW_NB15_testing-set.csv and UNSW_NB15_training-set.csv were already prepared as training and testing sets for the partition activity in this dataset. There are a total of 82,332 records in the TRAINING set and 175,341 records of various categories (normal and attack) in the TESTING set. In this section the results and Explanability analysis for decision tree, MLP, and XGBoost are presented.

3.1 Performance using Decision Tree Classifier

The tree function from the Scikit-learn library was used to implement a Decision Tree model on training data to sort behaviors as Attack or Normal. As shown in Figure 3, the model achieved an accuracy of 96.5% when evaluated against the testing set. The importance of a feature is determined by calculating the ratio of the reduction in node impurity to the probability of reaching that node. As a result, the most vital features will be positioned higher up in the tree. Since a single characteristic may be applied to several nodes, its relevance lies in the sum of the benefits it provides in terms of minimizing contamination.

```
[ ] cv_results['test_accuracy'].mean()
    0.9657241273842143
```

Figure 3: Decision Tree Classifier Report

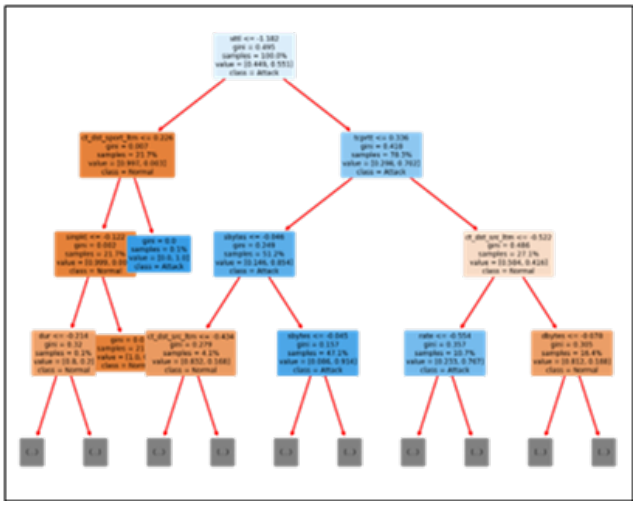

Figure 4: Decision Tree Classifier (Depth = 3 Nodes) Explainable AI Visualization

Both the scikit-learn library and the ELI5 Permutation significance toolkit were used to generate bar charts depicting the relative significance of the top 10 traits. The significance of a feature is calculated by comparing the reduction in node impurity to the probability of reaching that node. When visualized, the resulting representation will take the form of a tree, with the most important features positioned at the top.

The feature "source to destination time to live value" or "sttl" was shown to be the most essential to classification prediction in both feature significance outputs from the network traffic analysis. Figures 4 show a decision tree representation, with the most crucial characteristics shown in the higher levels.

3.1.1 Analysis of Decision Tree. When reporting cross-validation findings, we typically use the average of each statistic. The average accuracy score, for instance, may be determined by: each decision level and the related feature and splitting value for each condition may be inspected in the decision tree visualizations, making the model more explicable. If the condition is met by a given sample of network traffic, the sample is sent down the left node or branch; or else, it is sent down the right node. The forecast outcome of the classification is also shown in every class line, with the maximum depth of the tree being the determining factor. High accuracy classification findings from ML-based IDSs using decision trees for IoT network data indicate reliable detection of malicious threats. In addition, the decision tree (DT) algorithm's Explainability properties aid human analysts in comprehending the model. As a result, we can learn more about the state of cybersecurity in IoT networks. Theorizing what the IDS machine inferred from the characteristics or comparing expectations is also part of this comprehension. The subject expertise of a human analyst may be used to help a machine

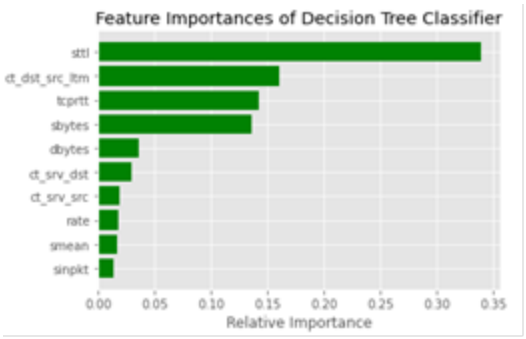

Figure 5: Importance of Features in Decision Trees using Scikit learn

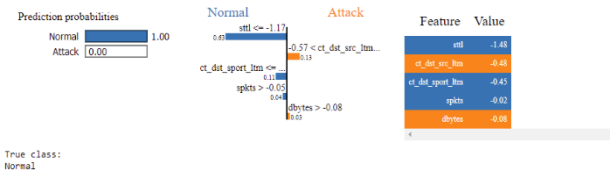

Figure 6: Single Classification Prediction using the MLP Classifier Explanation

learn by adding features or building features. This will be a huge aid to analysts as they attempt to evaluate and enhance the model decision framework's accuracy. Figure 5 shows the importance of features for decision tree classifier analysis.

3.2 Performance of Multi-layer Perceptron (MLP) Classifier

The MLP classifier was developed by subjecting the model to training and testing on the appropriate datasets. Overall, the model performed accurately 89.83% of the time when measured against the testing set.

This predicts a high success rate in identifying Normal or Attack behavior in IoT communications, indicating a very high classification prediction score. The LIME - Local Interpretable Model-Agnostic Explanations library can assist in visualizing the model predictions of the MLP Classifier for each prediction within the training set. LIME manipulates the attributes and predictions of the original data set, feeding it into an in-house classification model, as shown in Figure 6.

After then, the library gives more credence to fresh data outputs that are more closely associated with the starting point. Then, it uses the sample weights to fit a surrogate linear regression to the data. Finally, the newly trained explanatory model may be applied to the original data points.

3.3 XGBoost Classifier

The XGBoost Classifier, like the other two, was trained on the training set and was subsequently assessed on the testing set to verify its efficacy in data classification. Overall, the accuracy of the model was 89.89%, proving that the XGBoost classifier is quite good at identifying patterns in network behaviour. Similar to the performance of the MLP Classifier.

The SHAP (SHapley Additive exPlanations) package was used to provide this classifier explainability features. The SHAP library allows users to determine which characteristics and samples in training have the greatest effect on model or classifier performance. SHAP excels in tree-based model frameworks like XGBoost thanks to its local explanation and consistency. SHAP generates values for reading off of tree-based model output. It delivers granular feature significance via'marginal contribution to the model result,' which is based on value estimates from game theory. Figure 7 depicts an explanation for a single prediction. When analyzing network traffic records, the f(x) values offer a categorization value, with values closer to 1 indicating Attack behavior and values closer to 0 indicating Normal Activity.

4 CONCLUSIONS

In conclusion, this research investigated the performance of ML-based IDS and analyzed the explanability of the classifiers employed. The findings revealed that Decision Tree achieved an accuracy of 85%, Multilayer Perceptron (MLP) achieved an accuracy of 89.83%, and XGBoost achieved an accuracy of 89.9% in detecting attacks and normal network traffic. The analysis of explanability indicated that the Decision Tree classifier demonstrated better interpretability compared to MLP and XGBoost. However, it also exhibited lower accuracy in detecting intrusions. On the other hand, MLP and XGBoost classifiers showcased higher accuracy, but their explanability was relatively limited. These results highlight the trade-off between accuracy and interpretability in IDS using machine learning. While Decision Tree provides a clearer understanding of the decision-making process, MLP and XGBoost offer superior accuracy in identifying network intrusions. Therefore, the choice of classifier depends on the specific requirements and priorities of the system.

Overall, the study emphasizes the significance of machine learning in intrusion detection, with each classifier offering unique strengths and limitations. Further research and development in this area could focus on enhancing the interpretability of MLP and XGBoost classifiers without compromising their accuracy, ultimately leading to more effective and explainable IDS solutions.

REFERENCES

[1] A. Jannat, U. Hayat, and T. Sadiq, "Exploration of Machine Learning Algorithms for Development of Intelligent Intrusion Detection Systems," in 2023 International Conference on Communication, Computing and Digital Systems (C-CODE), Islamabad, Pakistan: IEEE, May 2023, pp. 1–6. doi: 10.1109/C-CODE58145.2023.10139885.

[2] S. Pontarelli, G. Bianchi, and S. Teofili, "Traffic-Aware Design of a High-Speed FPGA Network Intrusion Detection System," IEEE Trans. Comput., vol. 62, no. 11, pp. 2322–2334, Nov. 2013, doi: 10.1109/TC.2012.105.

[3] C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, and M. Rajarajan, "A survey of intrusion detection techniques in Cloud," J. Netw. Comput. Appl., vol. 36, no. 1, pp. 42–57, Jan. 2013, doi: 10.1016/j.jnca.2012.05.003.

[4] Nguyen Thanh Van, Tran Ngoc Thinh, and Le Thanh Sach, "An anomaly-based network intrusion detection system using Deep learning," in 2017 International Conference on System Science and Engineering (ICSSE), Ho Chi Minh City, Vietnam: IEEE, Jul. 2017, pp. 210–214. doi: 10.1109/ICSSE.2017.8030867.

[5] C. Chen, Yunchao Gong, and Yingjie Tian, "Semi-supervised learning methods for network intrusion detection," in 2008 IEEE International Conference on Systems, Man and Cybernetics, Singapore, Singapore: IEEE, Oct. 2008, pp. 2603–2608. doi: 10.1109/ICSMC.2008.4811688.

Figure 7: Visualize a single prediction with XGBoost SHAP

[6] N. Moustafa, B. Turnbull, and K.-K. R. Choo, "An Ensemble Intrusion Detection Technique Based on Proposed Statistical Flow Features for Protecting Network Traffic of Internet of Things," IEEE Internet Things J., vol. 6, no. 3, pp. 4815–4830, Jun. 2019, doi: 10.1109/JIOT.2018.2871719.

[7] K. A. P. Da Costa, J. P. Papa, C. O. Lisboa, R. Munoz, and V. H. C. De Albuquerque, "Internet of Things: A survey on machine learning-based intrusion detection approaches," Comput. Netw., vol. 151, pp. 147–157, Mar. 2019, doi: 10.1016/j.comnet.2019.01.023.

[8] A. Barredo Arrieta *et al.*, "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," Inf. Fusion, vol. 58, pp. 82–115, Jun. 2020, doi: 10.1016/j.inffus.2019.12.012.

[9] N. Moustafa and J. Slay, "The Significant Features of the UNSW-NB15 and the KDD99 Data Sets for Network Intrusion Detection Systems," in 2015 4th International Workshop on Building Analysis Datasets and Gathering Experience Returns for Security (BADGERS), Kyoto, Japan: IEEE, Nov. 2015, pp. 25–31. doi: 10.1109/BADGERS.2015.014.

[10] M. Sarhan, S. Layeghy, and M. Portmann, "Towards a Standard Feature Set for Network Intrusion Detection System Datasets," Mob. Netw. Appl., vol. 27, no. 1, pp. 357–370, Feb. 2022, doi: 10.1007/s11036-021-01843-0.

[11] M. Sarhan, S. Layeghy, N. Moustafa, and M. Portmann, "NetFlow Datasets for Machine Learning-Based Network Intrusion Detection Systems," in Big Data Technologies and Applications, Z. Deze, H. Huang, R. Hou, S. Rho, and N. Chilamkurti, Eds., in Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 371. Cham: Springer International Publishing, 2021, pp. 117–135. doi: 10.1007/978-3-030-72802-1_9.

[12] K. Amarasinghe, K. Kenney, and M. Manic, "Toward Explainable Deep Neural Network Based Anomaly Detection," in 2018 11th International Conference on Human System Interaction (HSI), Gdansk, Poland: IEEE, Jul. 2018, pp. 311–317. doi: 10.1109/HSI.2018.8430788.

[13] K. Siddique, Z. Akhtar, F. Aslam Khan, and Y. Kim, "KDD Cup 99 Data Sets: A Perspective on the Role of Data Sets in Network Intrusion Detection Research," Computer, vol. 52, no. 2, pp. 41–51, Feb. 2019, doi: 10.1109/MC.2018.2888764.

[14] S. Mane and D. Rao, "Explaining Network Intrusion Detection System Using Explainable AI Framework," 2021, doi: 10.48550/ARXIV.2103.07110.