IoT Network Intrusion Detection and Classification using Explainable (XAI) Machine Learning Algorithms

ISE 5194 – Human-Centered Machine Learning
Spring 2021

Harshil Patel

Abstract

The continuing increase of Internet of Things (IoT) based networks have increased the need for Computer networks intrusion detection systems (IDSs). Over the last few years, IDSs for IoT networks have been increasing reliant on machine learning (ML) techniques, algorithms, and models as traditional cybersecurity approaches become less viable for IoT. IDSs that have developed and implemented using machine learning approaches are effective, and accurate in detecting networks attacks with high-performance capabilities. However, the acceptability and trust of these systems may have been hindered due to many of the ML implementations being 'black boxes' where human interpretability, transparency, explainability, and logic in prediction outputs is significantly unavailable. The UNSW-NB15 is an IoT-based network traffic data set with classifying normal activities and malicious attack behaviors. Using this dataset, three ML classifiers: Decision Trees, Multi-Layer Perceptrons, and XGBoost, were trained. The ML classifiers and corresponding algorithm for developing a network forensic system based on network flow identifiers and features that can track suspicious activities of botnets proved to be very high-performing based on model performance accuracies. Thereafter, established Explainable AI (XAI) techniques using Scikit-Learn, LIME, ELI5, and SHAP libraries allowed for visualizations of the decision-making frameworks for the three classifiers to increase explainability in classification prediction. The results determined XAI is both feasible and viable as cybersecurity experts and professionals have much to gain with the implementation of traditional ML systems paired with Explainable AI (XAI) techniques.

1. Introduction

IoT networks have become an increasingly valuable target of malicious attacks due to the increased amount of valuable user data they contain. It has high significance to critical services where cyber-attacks attempt to compromise the security principles of confidentiality , integrity, and availability. In response, network intrusion detection systems (IDS) or systems that monitors and detects cyber-attack patterns over networking environments based on machine learning algorithms/models have been developed to detect suspicious network activity where They monitor network traffic for suspicious activities and issue alerts in case of detected attack types. Machine Learning (ML) algorithm are being investigated as potential IDS frameworks as current or traditional IDS capabilities have concerns including:

- IoT traditional network security solutions may not be directly applicable due to the differences in IoT structure and behavior.
- Low operating energy and minimal computational capabilities.
- Traditional security mechanism such as encryption protocols and authentication cannot be directly applied,
- Lack of a single standard for IoT architecture, policies, and connectivity domains.

Existing ML algorithms and models are able to learn IoT network inputs associated with target features concerning Normal or Attack behaviors to detect malicious activity or botnets. But current research indicates these ML-based IDSs are mainly "black boxes" where users of the systems in cybersecurity services are lacking the ability to explain why the system arrived at the prediction or classification of attack, which is important for optimal initial evaluation in cybersecurity and information assurance planning and resource allocation for IoT networks. The rapid integration of IoT networks in a variety of applications and setting has increased demand for robust cybersecurity practices and systems, but relevant researchers, security administrators, information security professional, etc…would benefit greatly from increased capabilities in understanding the ML-based IDS systems they may employ to effectively conduct operations and develop computer or network protection strategies to protect assets.

The objective of this work proposes to apply a survey of ML algorithms that have been modified and considered Explainable AI (XAI) methods through utilization of existing Python libraries to explain model classification decisions, logic behind predictions, and feature importance for predictability to increase transparency of ML-based IDSs to be understood further than current levels by human analysts in the IoT cybersecurity domain. These tools have been utilized in varying applications and show promise to extending explainability features to IoT network security IDS problems. Additionally, performance metrics of accuracy will be measured to augment explainability.

2. Literature Review

Protecting Internet of Things (IoT) networks has been a critical are of research for cybersecurity experts as IoT continues to be integrated in applications including home automation, smart cities, modern health systems, and advanced manufacturing. Furthermore, developing Intrusion detection systems based on machine learning techniques and other statistical feature learning algorithms has been researched and been put into application minimally. A study [8] conducted by Nour Mustafa developed an ensemble-based machine learning intrusion detection technique based on proposed statistical flow features for protecting network traffic of internet of things. In his proposed system, statistical flow features were generated from initial analysis of network features. Thereafter, an AdaBoost ensemble learning method was applied to three machine learning algorithms including decision tree, Naive Bayes (NB), and artificial neural network. The developed models were analyzed for performance to detect malicious events effectively, based on the UNSW-NB15 and NIMS botnet datasets with simulated IoT sensors' data. The experimental results conveyed high performance of detection of normal and malicious activity. Overall, the proposed ensemble technique has a higher detection rate and a lower false positive rate when compared to three other traditional IoT cybersecurity techniques. Many other machine learning-based intrusion detection systems have been researched and developed as outlined by Kelton da Costa survey [3]. His research investigates the range of machine learning techniques applied in Internet-of-Things and Intrusion Detection for computer network security by cybersecurity experts. Over 95 works on the subject were analyzed, that ranged across different sub-disciplines in machine learning and security issues in IoT environments.

Concerning literature on Explainable Artificial Intelligence (XAI), an extensive publication [4] summarizing the key concepts, taxonomies, opportunities, and challenges with aspect to

responsible AI can be used to review overall research work into explainability of ML methods. Recently, XAI had gained notable momentum, as lack of explainability has become an inherent problem of the latest ML techniques such as ensembles or Deep Neural Networks. Hence, XAI, has become crucial for deployments of ML models, where researchers keep transparency, fairness, model explainability and accountability at its core. More specifically, early investigation into developing Network Intrusion Detection System using an Explainable AI Frameworks [1] has conducted by Shraddha Mane and Dattaraj Rao. As ML models offer increased accuracy, the complexity increases and hence the interpretability decreases. In their paper, they developed a deep neural network for network intrusion detection and proposed an explainable AI framework to demonstrate model transparency throughout the machine learning pipeline. Utilizing existing XAI algorithms generated from SHAP, LIME, Contrastive Explanations Method (CEM), Protidic and Boolean Decision Rules via Column Generation (BRCG), that provide explanations on individual predications, they applied the approaches to the NSL-KDD dataset demonstrating successful increase in model transparency.

3. Overview & Benefits of Explainable (XAI) Machine Learning

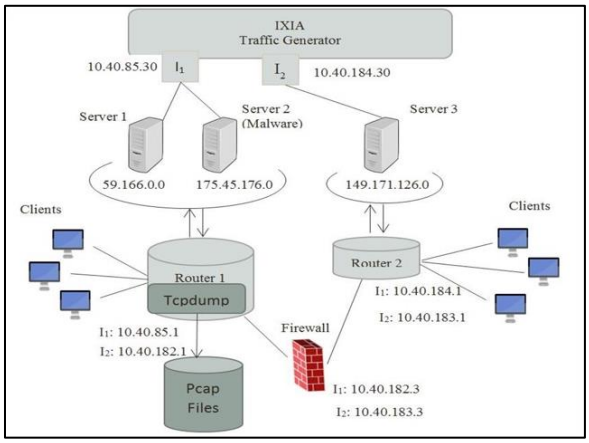
Advanced and state of the art ML algorithms and models offer valuable applications in establishing better IoT network security. The ML techniques, learn from input features generated in network traffic, and offer support to cybersecurity personnel in making critical threat detection decisions. However, these techniques are based on advanced models that are too complex to be interpreted by human analysts; hence, may they turn to traditional tools that may not be as viable but offer more explainability or inherent trust by the human involved. In many cases, it is nearly impossible to get a feeling for its inner workings of a ML system for Intrusion Detection. This may further decrease trust that a certain prediction from the model is correct even though performance results may indicate otherwise. Having an intuitive explanation of the rationale behind individual predictions or model decision-making framework will better position cybersecurity experts to trust prediction or the classifiers itself, especially, in understanding how it behaves in particular cases. Explainable AI (XAI) offers a variety of explanation or feature importance tools for generating explanations about the knowledge captured by trained ML models to aid in increasing overall trust.

4. Methodology

4.1 UNSW-NB15 Dataset

UNSW-NB15 is an IoT-based network traffic data set with different categories for normal activities and malicious attack behaviors from botnets (through classification of attack type including Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode and Worms). The raw network packets of the UNSW-NB 15 dataset were created by the IXIA PerfectStorm tool in the Cyber Range Lab of the Australian Centre for Cyber Security (ACCS) for generating a hybrid of real modern normal activities and synthetic contemporary attack behaviors on IoT based networks. Figure 1 shows the testbed configuration dataset and the method of the feature creation of the UNSW-NB15.

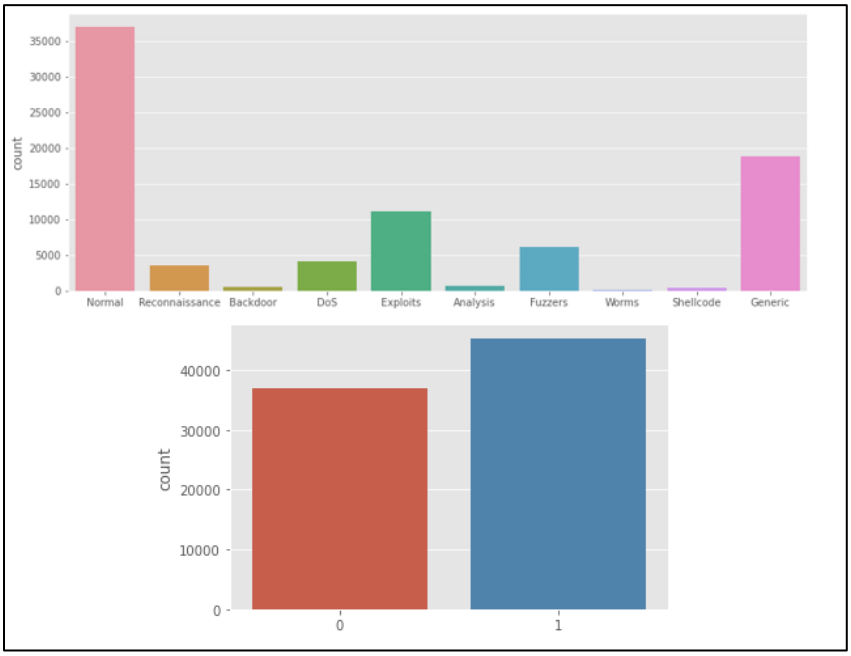
Figure 1: IXIA Traffic Generator Overview:



The UNSW-NB15 is pre-partitioned by its creators into being configured into a training set for model training and testing set for model performance, namely, UNSW_NB15_training-set.csv and UNSW_NB15_testing-set.csv respectively. The number of records in the training set is 175,341 records and the testing set is 82,332 records with a target response of the traffic behavior for each record, attack, and normal behavior. The dataset consists of 39 features that are numeric in nature. The features and their descriptions are listed in the UNSW-NB15_features.csv file. The balance for the

For our experimental processes, the target feature will be a binary classification of Normal or Attack behavior. Figures 2 provide the details and the values distribution of each attack class within the data subsets, where 0 represents Normal and 1 represents Attack behavior. We could see that the dataset is adequately balanced for the binary response variable of activity behavior.

Figure 2: Training Dataset Distribution and Counts



4.2 Overview of ML Methods

The three supervised ML approaches that will be used develop binary classification classifiers are Decision Trees, Neural Network based on Multi-layer Perceptron, and XGBoost. The ML algorithms in the mentioned order offer decreasing capabilities for explainability (XAI).

Decision Tree Classifier

The decision tree (DT) classifier is a supervised ML algorithm that will be utilized for the classification task of Normal or Attack behavior based on the 39-input feature. The resulting DT algorithm develops a decision-making process based on a tree-like model with nodes or branches. The max depth of the decision tree can be defined beforehand. A decision tree is already an explainable machine learning algorithm through visualizations of the resulting trees.

Multi-layer Perceptron Classifier

Multi-Layer Perceptron(MLP) classifiers are artificial neural networks which utilizes perceptrons or single neuron models for complex predictive modeling tasks. The MLP classifier as a neural net have an inherent ability to learn the representation in your training data and how to best relate it to the output variable that you want to predict. In this sense neural networks learn a mapping. The building block for neural networks are artificial neurons, which weighted input signals and produce an output signal using an activation function. The activation function is a mapping of summed weighted input to the output of the neuron and contains the threshold at which the neuron is activated and strength of the output signal. Neurons are arranged into layers of neurons, and layers creates a network, where the weights initially are best guesses. The most preferred way training algorithms for neural networks are gradient descent or back-propagation algorithm with sigmoid function, in which the network processes the input upward activating neurons as it goes to finally produce an output value. The output of the network is compared to the expected output and an error is calculated. This error is then propagated back through the network, one layer at a time, and the weights are updated according to the amount that they contributed to the error. The process is repeated for all of the examples in your training data, as the weights converge to a local optimum.

Neural networks like MLP Classifiers for the most part, lack sufficient model explainability and interpretability. In the tradeoff between the explainability/interpretability of an algorithm and its accuracy in application, neural networks heavily lean more toward the prediction performance. Neural networks contain visible layers and hidden layers of neural units, which hidden layers and its unknown interaction post training significantly causes neural networks to act as “black-box” algorithms instead.

XGBoost Classifier

XGBoost is an implementation of gradient boosted decision trees designed for speed and performance. XGBoost stands for ‘eXtreme Gradient Boosting’. XGBoost is provided as an open-source software library where implementation of the algorithm was engineered for efficiency of compute time and memory resources. The design architecture allows for best use of available resources to train the model. The XGBoost library implements a gradient boosting decision tree algorithm. Boosting is an ensemble technique in which new models are added to correct the errors

made by existing models. Gradient boosting is an approach where new models are created that predict the residuals or errors of prior models and then added together to make the final prediction. Furthermore, the implemented gradient descent algorithm minimizes the loss when adding new models. This approach will support the classification predictive modeling of Normal or Attack behavior.

4.3 Proposed Approach with Scikit-learn, XGBoost, and XAI Libraries

UNSW-NB15 training dataset after applying data processing techniques for data cleaning, normalization, and transformation will be used to train each of the three supervised ML binary classifiers: Decision Trees, Neural Network based on Multi-layer Perceptron, and XGBoost. The target feature will be a binary classification of Normal (0) or Attack (1) behavior. Thereafter, the next process will be to test the trained model using the data processed UNSW-NB15 testing dataset. The model performance will be evaluated using the accuracy score. The procedure described above is will not be tuned using model or classifier hyperparameters. Scikit-Learn implementation of the Decision Trees Classifier and Multi-layer Perceptron Classifier will be utilized, while the XGBoost library will be utilized for the XGBoost Classifier.

After classifiers are trained and tested, the next process is to develop interpretable diagrams, feature importance plots, and classification/prediction explanation visuals based on the trained classifiers used to detect network traffic behavior in the testing set. The following Python packages are and investigate to modify the ML classifiers for explainability:

1.) **ELI5** is a visualisation library that is useful for debugging machine learning models and explaining the predictions they have produced.

2.) **LIME (local interpretable model-agnostic explanations)** is a package for explaining the predictions made by machine learning algorithms.

3.) **SHAP (SHapley Additive exPlanations)** is a game-theoretic approach to explaining the output of any machine learning model. SHAP to help better understand the impact of features on the model output.

5. Results & Discussion

Decision Tree Classifier

Using the Scikit-learn library's tree.DecisionTreeClassifier(), the training set was used to build a Decision Tree classification model for Normal or Attack behavior. The model performance accuracy against the testing set was 85% as indicated in Figure 3.

Figure 3: Decision Tree Classifier Report

```
Accuracy: 0.8510901614568184
Reporting for ['Decision Tree Classifer', 'RegLog']:
              precision    recall  f1-score   support

           0       0.69      0.98      0.81     56000
           1       0.99      0.79      0.88    119341

    accuracy                           0.85    175341
   macro avg       0.84      0.88      0.84    175341
weighted avg       0.89      0.85      0.86    175341
```

The feature importance for the top 10 features was graphed with both the scikit-learn library and ELI5's Permutation Importance toolkit. Feature importance is calculated as the decrease in node impurity weighted by the probability of reaching that node. The most important features will be higher in the tree-like visualization generated.

Figure 4: Decision Tree Feature Importance: Scikit Learn

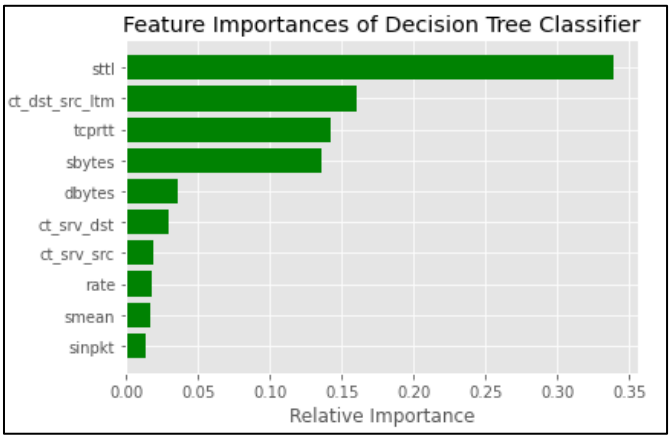


Figure 5: Decision Tree Feature Importance: ELI5 Permutation Importance

Weight	Feature
0.2755 ± 0.0003	sttl
0.2411 ± 0.0007	ct_dst_sport_ltm
0.1359 ± 0.0014	sbytes
0.0707 ± 0.0012	ct_dst_src_ltm
0.0354 ± 0.0002	sloss
0.0288 ± 0.0009	smean
0.0108 ± 0.0005	dbytes
0.0011 ± 0.0001	sinpkt
0.0005 ± 0.0001	ct_dst_ltm
0 ± 0.0000	sjit
0 ± 0.0000	dinpkt
0 ± 0.0000	dpkts
0 ± 0.0000	dloss
0 ± 0.0000	stcpb
0 ± 0.0000	swin
0 ± 0.0000	sload
0 ± 0.0000	rate
0 ± 0.0000	dload
0 ± 0.0000	djit
0 ± 0.0000	is_sm_ips_ports
… 19 more …	

Both of the feature importance outputs indicate very similar results with feature ‘sttl’ or “source to destination time to live value” in the network traffic analysis being indicated as the most important to classification prediction. The most important features can be visualized in the upper layers of the decision tree visualization in Figures 6 through 8.

Figure 6: Decision Tree Classifier (Depth = 3 Nodes) Explainable AI Visualization

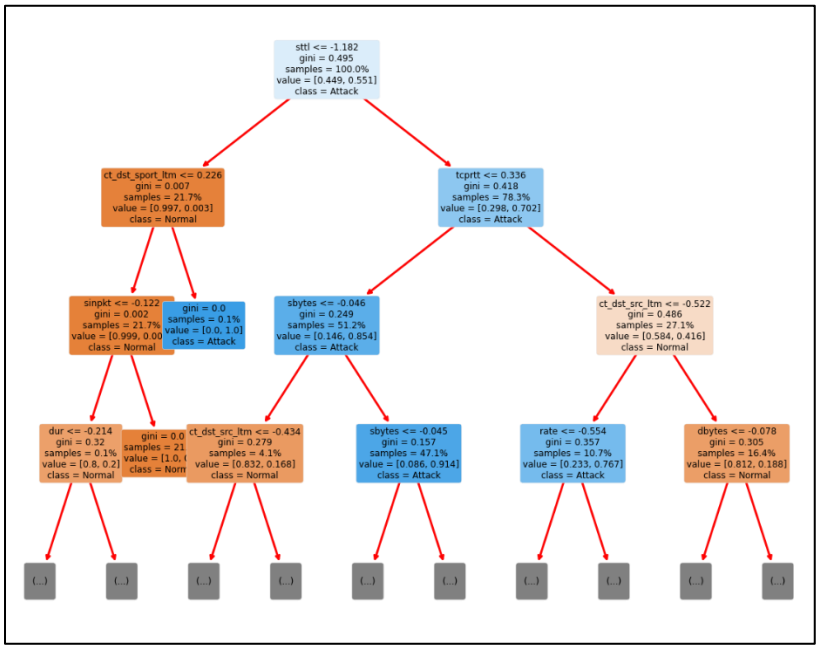


Figure 7: Decision Tree Classifier (Depth = 5 Nodes) Explainable AI Visualization

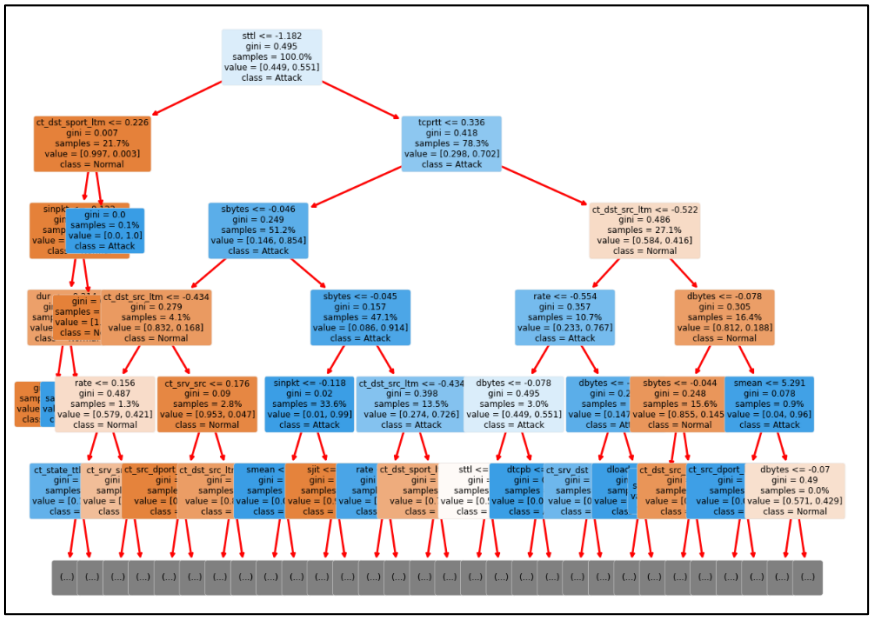
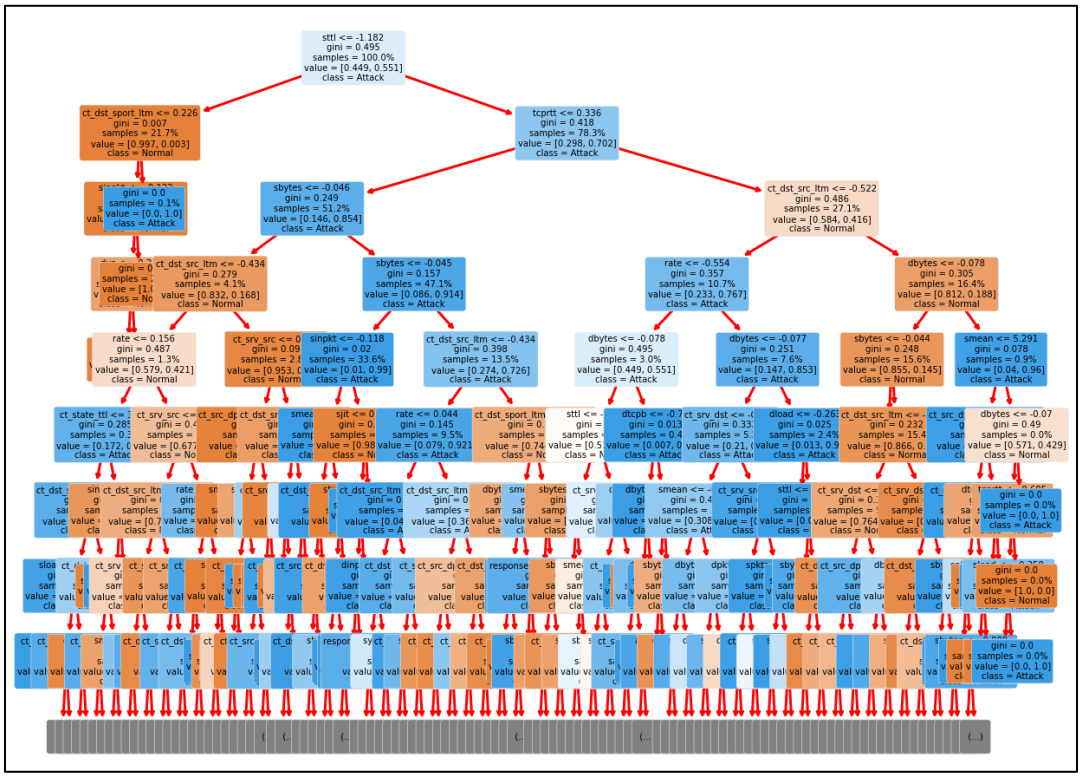


Figure 8: Decision Tree Classifier (Depth = 8 Nodes) Explainable AI Visualization



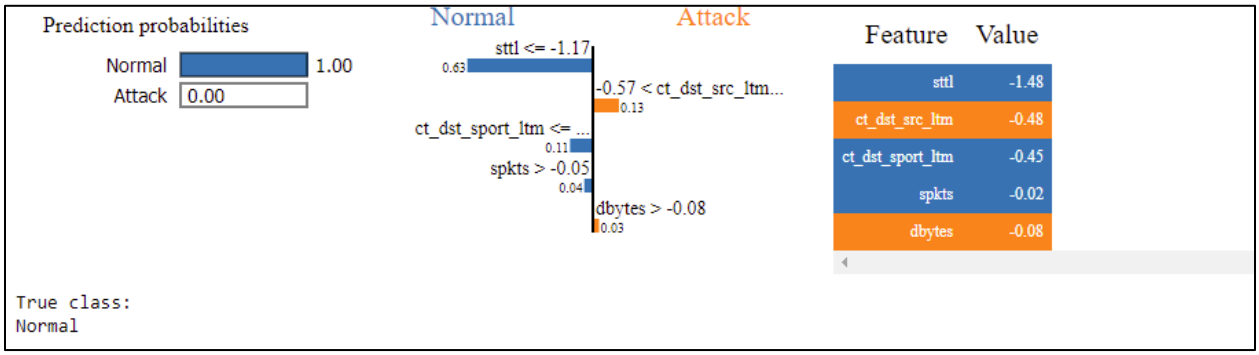
The decision tree visualizations enables model explainability through inspection of each decision level and its associated feature and splitting value for each condition. If a certain network traffic sample satisfies the condition, it goes to the left branch or node, otherwise it goes to the right branch. Additionally, in each class line, the classification prediction result is depicted depending on the max depth of the tree selected. Utilizing decision trees for IoT network traffic ML-based IDSs provide high accuracy classification results, indicating robust detection of malicious threats. Furthermore, the explainability features of the DT algorithm based on plotting decision trees can help human analysts understand the model. This will allow for greater understanding of the cybersecurity landscape around IoT networks. This understanding includes theorizing what the IDS machine learned from the features or comparing expectations. Human analyst may further aid the machine in learning though adding features or feature engineering using domain knowledge. This will significantly help analysts assess the correctness of the model decision framework and improve upon it.

Multi-layer Perceptron (MLP) Classifier

For the MLP classifier, the model was trained and tested on the corresponding datasets. The overall model performance accuracy against the testing set was 89.83%. This indicates a very exceptional classification prediction score of detecting Normal or Attack behavior in IoT traffic. Using the LIME - Local Interpretable Model-Agnostic Explanations library, a model predication visual of the MLP Classifier can be generated for individual predications in the training set. LIME perturbs the original data features and prediction, to feed into a developed internal classification model, and then observes the outputs. Thereafter, the library weighs the new data outputs as a function of their

proximity to the original point. Next, it fits a surrogate linear regression on the dataset with variations suing the sample weights. Lastly, the original data points can be explained with the newly trained explanation model. Figure 9 displays an example of the Lime Tabular Explainer output with the top 5 features indicated.

Figure 9: Single Classification Prediction using the MLP Classifier Explanation



The visual dashboard indicate which features and their weights brought the overall behavior classification to be predicted as Normal for that network traffic record. This classification is inspected to be correct as the true class is 'Normal'. This visual dashboard offers robust individual explainability of predicted classifications. Human analysts can conduct in-depth analysis for cybersecurity research or follow-up assessment on why certain network traffic was classified in which they were by the model. This tools offers increased transparency capability of predictions that can be exploited for future cybersecurity research, while utilizing the high-performance benefits of a neural net MLP classifier, which are functionally 'black boxes.

XGBoost Classifier

Similarly, to the other two classifiers, the XGBoost Classifier was trained for the classification task and tested on the testing set. The overall model performance accuracy was 89.89%, demonstrating high capability of the XGBoost classifier to classify network behavior. The performance is approximately similar to the MLP Classifier.

To utilize explainability capabilities with this classifier, the SHAP (SHapley Additive exPlanations) library was utilized. The SHAP library offers the ability to analyze which training samples and features offer the highest impact on model or classifier output. SHAP's main advantages are local explanation and consistency in tree-based model structures such as XGBoost. SHAP creates values that interpret results from tree-based models. It is based on value calculations from game theory and provides extensive feature importance using by 'marginal contribution to the model outcome'.

To explain predictions, the Tree SHAP implementation integrated into XGBoost can be used to explain the testing set classification predictions. Figure 10 provides a visualization into explaining single prediction, while Figure 11 captures an explanation into many predictions through feature comparison or output classification values. The f(x) values provides a classification value, where closer to 1 indicates Attack behavior, while closer to 0 indicates Normal Activity by a network traffic record.

Figure 10: XGBoost SHAP- Visualize a single prediction

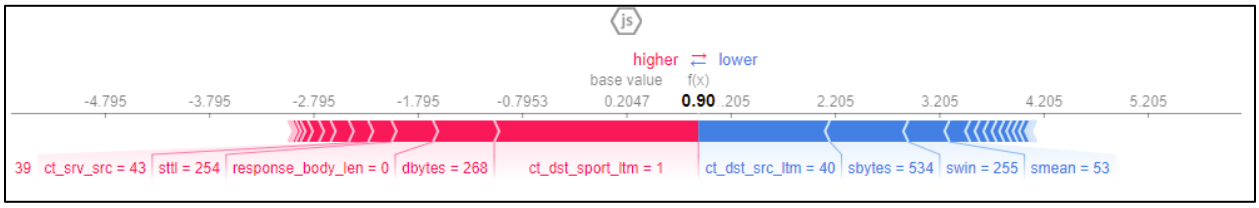
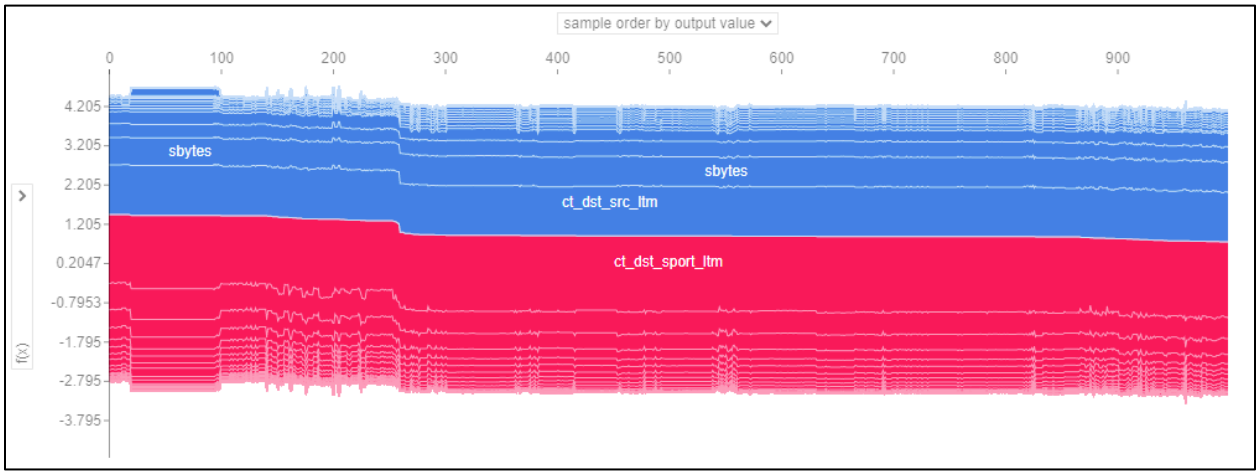
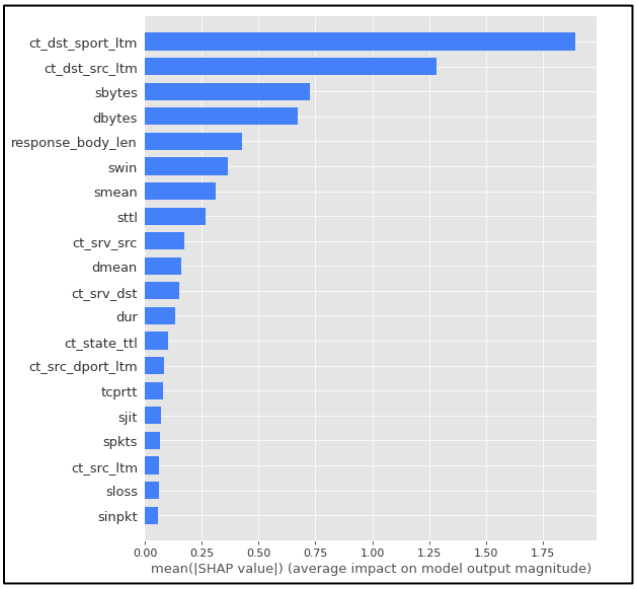


Figure 11: XGBoost SHAP - Visualize many predictions



A feature importance plot through SHAP is conveyed in Figure 12 to determine the mean importance of input training features to predict classification. The results are similar to the DT Classifier.

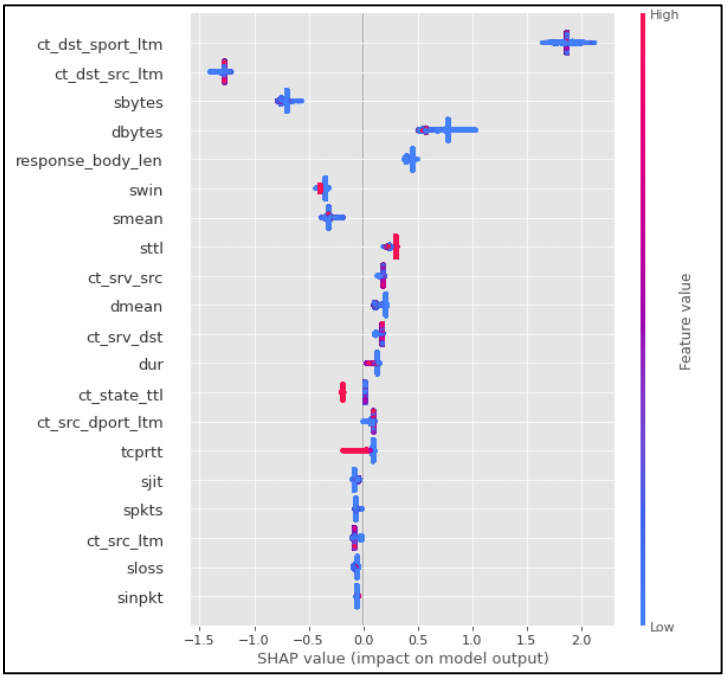
Figure 12: SHAP Feature Importance on XGBoost Classifier



The SHAP summary plot shows the top feature combinations, while providing visual indicators of how feature values affect classification predictions. In Figure 13, red indicates higher feature value, and blue indicates lower feature value. On the x-axis, higher SHAP value to the right

corresponds to prediction value (Attack Behavior), lower SHAP value to the left corresponds to lower prediction value (Normal Activity).

Figure 13: SHAP Summary Plot for XGBoost



Furthermore, the SHAP values can create SHAP dependence plots, which show the effect of a single feature across the whole dataset. They plot a feature's value vs. the SHAP value of that feature across many samples and account for interaction effects present in the features. Additionally, a summary plot of a SHAP interaction value matrix plots a matrix of summary plots with the main effects on the diagonal and the interaction effects off the diagonal.

Figure 13: SHAP Dependence Plots for ‘sttl’ feature

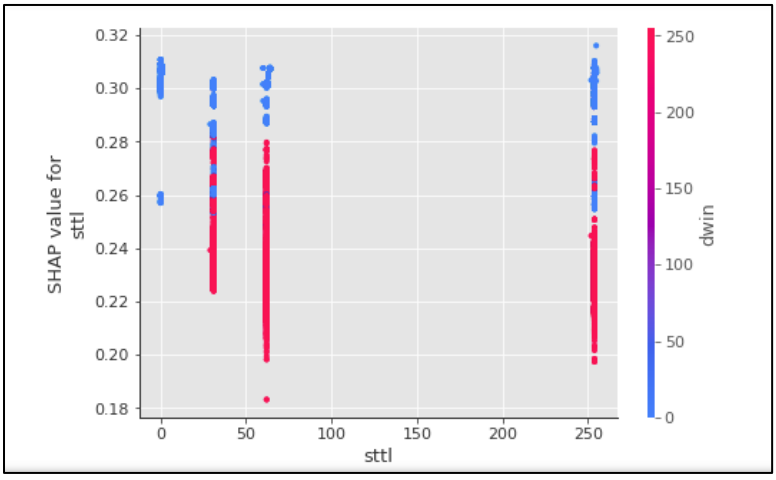
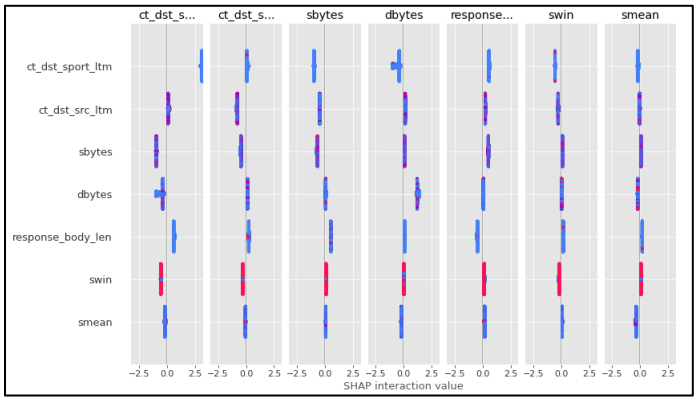
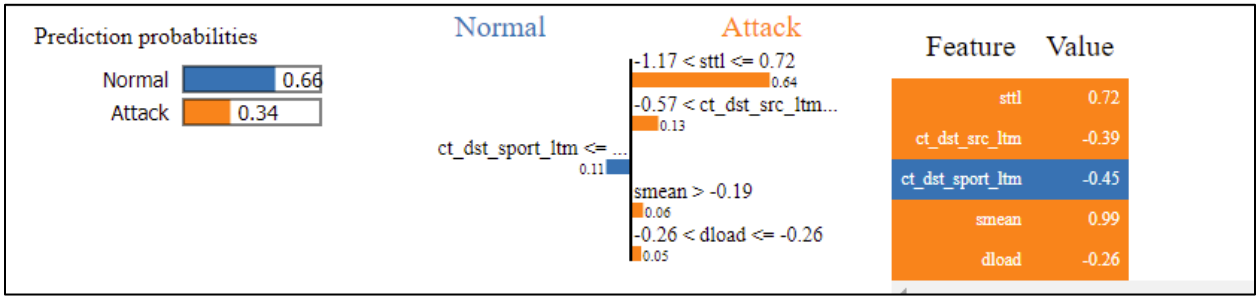


Figure 14: SHAP Interaction Value Summary Plot



Furthermore, using the LIME package as used for the MLP Classifier, individual predictions of the XGBoost Classifier can be explained.

Figure 15: Single Classification Prediction using the XGBoost Classifier Explanation



The model offers a highly efficient and flexible, while high-performing classifier that can be paired with the SHAP and LIME libraries offer robust explainability features. This will increase trustworthiness of advanced black-box algorithms for effective evaluations of ML-based IDSs for IoT network security.

6. Conclusion

ML learning models utilized for IoT network traffic security through IDSs are increasing becoming more complex, but the need for human analysts to analyze outcomes through inherent domain knowledge for resource allocation and cybersecurity strategy development is a critical role. ML algorithms are often considered “black boxes”, in which the logic or explanation behind the output predictions are not interpretable. Through utilizing the UNSW-NB15 dataset and training a Decision Tree Classifier, MLP Classifier, and XGBoost Classifier, the accuracy results conveyed high-performance for analyzing network behavior of Attack or Normal Activity between connected clients in a IoT network. After, analyzing the performance of ML classifiers, established libraries and techniques for enabling explainability or Explainable AI (XAI) were applied to the trained classifiers to explain its decisions and evaluate feature importance. In the immediate term, this increased transparency will increase trust with ML systems in the IoT cybersecurity domain.

Ultimately, it will enable a new range of capabilities of IoT cybersecurity trough extracting insights from sophisticated machine learning models as more explainability conveys the influence of the influence the prediction of a cyber-attack and to what degree.

7. Acknowledgments

The datasets used during the current study are available at <https://www.kaggle.com/mrwellsdavid/unsw-nb15/code>.

This current study builds upon previous work for Dr. Theodore Allen's course at the Ohio State University, ISE 5194: Introduction to Operations Analytics in Spring 2020. This work was limited to a binary classification project that was conducted utilizing multi-linear regression and logistic regression on the UNSW-NB15 for behavior type prediction and direct classification of attack type.

The overall files outlining all activities conducted is available as .ipybn and .py file formats and is attached to this publication. Additionally, it is available at <https://github.com/harshilpatel1799/IoT-Network-Intrusion-Detection-and-Classification-using-Explainable-XAI-Machine-Learning>.

8. References

1. Mane, Shraddha, and Dattaraj Rao. "Explaining Network Intrusion Detection System Using Explainable AI Framework." arXiv preprint arXiv:2103.07110 (2021).
2. Moustafa, Nour, and Jill Slay. "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)." Military Communications and Information Systems Conference (MilCIS), 2015. IEEE, 2015.
3. da Costa, Kelton AP, et al. "Internet of Things: A survey on machine learning-based intrusion detection approaches." Computer Networks 151 (2019): 147-157.
4. Arrieta, Alejandro Barredo, et al. "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI." Information Fusion 58 (2020): 82-115.
5. Zoghi, Zeinab, and Gursel Serpen. "UNSW-NB15 Computer Security Dataset: Analysis through Visualization." arXiv preprint arXiv:2101.05067 (2021).
6. García-Magariño, R. Muttukrishnan and J. Lloret, "Human-Centric AI for Trustworthy IoT Systems With Explainable Multilayer Perceptrons," in IEEE Access, vol. 7, pp. 125562-125574, 2019, doi: 10.1109/ACCESS.2019.2937521.
7. Wang Z: Deep learning-based intrusion detection with adversaries. IEEE Access. 2018;6:38367–384.
8. Moustafa, Nour, et al. "An Ensemble Intrusion Detection Technique based on proposed Statistical Flow Features for Protecting Network Traffic of Internet of Things." IEEE Internet of Things Journal (2018).

9. C. S. W. M. M. Daniel L. Marino, "An Adversarial Approach for Explainable AI in Intrusion Detection Systems," in IECON 2018 - 44th Annual Conference of the IEEE Industrial
10. K. Z. Y. Y. X. W. Maonan Wang, "An Explainable Machine Learning Framework for Intrusion Detection System," IEEE Access , vol. 8, pp. 73127 - 73141, 16 April 2020.
11. Koroniotis, Nickolaos, Moustafa, Nour, et al. "Towards Developing Network Forensic Mechanism for Botnet Activities in the IoT Based on Machine Learning Techniques." International Conference on Mobile Networks and Management. Springer, Cham, 2017.
12. Moustafa, Nour, et al. "A New Threat Intelligence Scheme for Safeguarding Industry 4.0 Systems." IEEE Access (2018).
13. Cohen, "Explainable AI (XAI) with a Decision Tree," *Medium*, 20-Apr-2021. [Online]. Available: https://towardsdatascience.com/explainable-ai-xai-with-a-decision-tree-960d60b240bd. [Accessed: 30-Apr-2021].
14. Kasongo, S.M., Sun, Y. Performance Analysis of Intrusion Detection Systems Using a Feature Selection Method on the UNSW-NB15 Dataset. J Big Data 7, 105 (2020). https://doi.org/10.1186/s40537-020-00379-6
15. M. T. Ribeiro, "LIME - Local Interpretable Model-Agnostic Explanations," LIME - Local Interpretable Model-Agnostic Explanations – Marco Tulio Ribeiro –. [Online]. Available: https://homes.cs.washington.edu/~marcotcr/blog/lime/. [Accessed: 30-Apr-2021].
16. "LIME: Local Interpretable Model-Agnostic Explanations," C3 AI, 19-Oct-2020. [Online]. Available: https://c3.ai/glossary/data-science/lime-local-interpretable-model-agnostic-explanations/. [Accessed: 30-Apr-2021].
17. M. Ribeiro, Tutorial - continuous and categorical features. [Online]. Available: https://marcotcr.github.io/lime/tutorials/Tutorial%20-%20continuous%20and%20categorical%20features.html. [Accessed: 30-Apr-2021].
18. S. Slundberg, "SHAP (SHapley Additive exPlanations)," slundberg/shap. [Online]. Available: https://github.com/slundberg/shap. [Accessed: 30-Apr-2021].