Attack and Anomaly Detection in IoT networks using Machine Learning

PROJECT REPORT

For

Machine Learning (CSE6005)

Slot: A2

Submitted By:

Name	Registration Number
HARPAL KAUR DHINDSA	20MCB0008
RAKSHITH VIKRAMRAJ	20MCB0009

Under the guidance of

DR. SAIRABANU J

(SCHOOL OF COMPUTER SCIENCE AND ENGINEERING)

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

June 2021

ATTACK AND ANOMALY DETECTION IN IOT NETWORKS USING MACHINE LEARNING

by Harpal Kaur Dhindsa, and Rakshith Vikramraj

1. ABSTRACT

Web traffic recognition and classification has been meticulously studied somewhat recently, yet this is as yet a hot point with respect to the Internet of Things (IoT), a communication platform that will potentially include various aspects of our day-to-day activities. Intrusion and anomaly detection in the Internet of Things (IoT) framework is a rising concern in the space of IoT. With the expanded utilization of IoT foundation in each domain, attacks and threats in these systems are additionally growing likewise. Malicious Control, Data Type Probing, Malicious Operation, Denial of Service, Scan, Spying and Wrong Setup are such attacks and irregularities which can potentially cause failure of the IoT framework. In this paper, exhibitions of a few Machine Learning models have been contrasted to predict attacks and anomalies on the IoT frameworks precisely. The machine learning (ML) algorithms that have been used here are Support Vector Machine (SVM), Isolation Forest (IF), Kernel Density Estimation (KDE), and Local Outlier Factor (LOF).

Keywords – Anomaly Detection, IoT Networks, Local Outlier Factor, Isolation Forest, Kernel Density Estimation, Machine Learning, Intrusion Detection.

2. INTRODUCTION

Outliers or anomalies are specific data patterns are data with different characteristics compared to the normal data instances. Web traffic recognition and classification has been meticulously studied somewhat recently, yet this is a hot point with respect to the Internet of Things (IoT). Intrusion and anomaly detection on the Internet of Things (IoT) framework is a rising concern in the space of IoT . With the expanded utilization of IoT foundation in each domain, attacks and threats in these systems are additionally growing likewise. Malicious Control, Data Type Probing, Malicious Operation, Denial of Service, Scan, Spying and Wrong Setup are such attacks and irregularities. The sensors in these IoT networks which generate the data and these generated data is high in volume due to large number of sensors and IoT modules spread across geographical location. These data are very important for many decision making policies. Thus the reliability of any data is most important in any application the sensors have limited resources and capability thus making the data generated by the sensors often unreliable and inaccurate. These sensors are battery operated and when these batteries get replenished the possibility of getting erroneous data increases by many folds environmental effects also play a major role in the operation of sensor nodes be it in IoT or WSN (Wireless Sensor Networks) the main aim is the data communication between devices without any human intervention thus the sensors are also susceptible to malicious attacks in which the data are exploited by adversaries all these factors contributed to unreliability of the sensor data which ultimately influence the final decision making process.

In this project we have used UNSW-NB15 Dataset, UNSW-NB15 is an IoT-based network traffic data set with different categories for normal activities and malicious attack behaviours (Jatti & Kishor Sontif, 2019)**.** we are proposing three methods to detect outliers in the IoT network which are Isolated Forest (IF) (F. T. Liu et al., 2008) , One-class SVM and Local Outlier Factor (LOF). These methods was tested against the UNSW-NB15 dataset and a comparative study has been made and the methodologies and result will be discussed in upcoming sections.

3. CONTRIBUTION OF WORK

There has been past implementations in which different strategies have been utilized to distinguish the anomaly in IoT Networks like parametric techniques (Logistic Regression Theory), non-parametric techniques (grey's system theory). Indeed, supervised learning techniques have been utilized, for example, support vector machine (SVM), K-nearest Neighbours, Neural Networks, Principle Component Analysis (PCA) and Fuzzy logics. Despite the fact that parametric techniques are basic and they have the capacity of catching on quickly from the information, they experience the ill effects of the downside of deduced suspicion of the information appropriation. As opposed to this, non-parametric strategies as examined before don't expect anything about the data distribution making it more malleable for any sort of data. In any case, they require a great deal of training data to make a mapping function. Classification based techniques generally give an exceptionally exact classification of anomalies by making a classification model. However, one of the fundamental downsides of SVM is its computational intricacy and picking the legitimate portion of data. Likewise, for Bayesian models if the quantity of factors is enormous, making an exact model stances a significant test.

In our work we have used UNSW-NB15 Dataset which is provided by UNSW Sydney, UNSW-NB15 is an IoT-based network traffic data set with different categories for normal activities and malicious attack behaviours (Jatti & Kishor Sontif, 2019)**.** we are proposing three methods to detect outliers in the IoT network which are Isolated Forest (IF) (F. T. Liu et al., 2008) , One-class SVM and Local Outlier Factor (LOF). These methods was tested against the UNSW-NB15 dataset and a comparative study has been. The result obtained is then visualized and the conclusion is carriedout on the basis of different parameter metrics, the main factor which was taken into considerations was F1 Score.

4. RELATED WORKS

In this paper the authors (Maniriho et al., 2020) proposes a HFS-Engine which is designed with two main feature selections modules: relevant feature subset module, which is used to select relevant features, and a feature selection merging module that employs the concept of mathematical sets (intersection rule) to select the most relevant features which are fed to the machine learning classifier.

In this paper the authors (Z. Liu et al., 2020) used machine learning approaches which are applied on the dataset for real-time and accurate anomaly detection. The approaches include Logistic Regression (LR), Support Vector Machine (SVM), K-nearest Neighbours (KNN), Random Forest (RF), and XG Boost. Binary Classification is applied by classifying anomalies as 0 and normal packet as 1. Overall, even though the RF approach gives the highest metric scores, it takes the highest computational effort.

In this paper the authors (Sahu & Mukherjee, 2020) have used two classification algorithms, they are Logistic Regression, and Artificial Neural Network and there was a significant accuracy increase in detecting outliers.

The authors (Yahyaoui et al., 2021) proposed a Reliable Event and Anomaly Detection Framework for the Internet of Things (READ-IoT for short). The proposed approach depends on two cascading stages enactment of detection parts.

The authors (Doshi et al., 2021) proposed a novel modification for ODIT, and prove that the said modified version, asymptotically becomes the Cumulative Sum (CUSUM) test as the training data size grows, which is the optimum sequential change detection algorithm in the minimax sense. CUSUM is a parametric test which accepts both the ostensible and the peculiar dispersions are totally known.

In this paper (Sadreazami et al., 2018) proposed a novel intrusion detection method based on changed Bayesian probability proportion test

In this paper the authors (Cristiani et al., 2020) proposed a Intrusion Detection System called FROST is used, that applies machine learning techniques, along with the fuzzy clustering, to identify and prevent different types of cyber-attacks in IoT environments. In addition, FROST has a mechanism for filtering and detecting anomalies, allowing the identification of new types of attacks, obscure to the framework, that may arise over time.

In this paper the authors (Chkirbene et al., 2020) proposed a hybrid anomaly-based intrusion detection system is projected by combining two supervised machine learning algorithms which are Random Forest, and Classification and Regression Trees (CART).

In this paper the authors (Hwang et al., 2020) proposed a CNN-based deep learning approach is proposed for auto-learning the traffic features and profiling traffic directly from the raw traffic with only a few first packets per flow. Following this, the auto-learning approach can fundamentally save the endeavours to construct traffic patterns for a mind-boggling network where the variety of use traffic is the significant test to regular strategies.

In this paper the authors (Gu et al., 2020) proposed to Incorporate reinforcement learning to develop a novel attack detection framework in IoT networks. Their approach can provide smarter defences with less human supervision, which can greatly facilitate the current research in IoT security.

In this paper the authors (Moussa & Alazzawi, 2020) proposed a novel to model a Cyber-Attacks Detection System. It is an Intrusion Detection System (IDS) implemented as a countermeasure against many types of cyber-attacks. Such a system is concerned with the detection of hostile actions.

In this paper the authors (Zidi et al., 2018) proposed a SVM classification method which uses the concept of hyperplane to differentiate between two classes. To determine the hyperplane, SVM maximizes the margin and tries to distinguish between the classes with minimal errors.

In this paper the authors (Xie et al., 2013) proposed In KNN, prediction for a new observation is made by searching through k nearest neighbours based on any distance measures, the most popular being the Euclidean distance. Based on the KNN cluster values the outliers are detected here.

In this paper the authors (Hodo et al., 2017) proposed a multi-layer perceptron ANN which is used analyse the intrusion in the system, two type of algorithms is used which is Feed forward Learning Algorithm and Backward Learning algorithm.

5. EXISTING SYSTEM DESCRIPTION

A group based ML interruption discovery procedure has been executed, in light of measurable flow features for securing network traffic of web of things. The statistical features are produced from starting examination of organization highlights. From that point, an AdaBoost gathering learning strategy was applied to three ML calculations including decision tree, Naive Bayes (NB), and artificial neural network. The created models were investigated for execution to recognize malicious activities successfully, in view of the UNSW-NB15 and NIMS botnet datasets consisting IoT sensors' information. The trial results passed on high performance for detection of ordinary and malicious action. Generally, the ensemble procedure has a higher

identification rate with lower false positive rate when contrasted with three others conventional IoT network safety strategies. Numerous other ML based interruption identification frameworks have been explored and created.

Concerning research on Explainable Artificial Intelligence (XAI), a broad distribution summing up the key ideas, scientific categorizations, applicability, and difficulties with viewpoint to an capable AI can be utilized to audit in general overall literature into explain ability of ML strategies. As of late, XAI had acquired striking force, as absence of explain-ability has become an inborn issue of the most recent ML strategies like clustering or Deep Neural Networks. Consequently, XAI, has gotten vital for arrangements of ML models, where scientists keep straightforwardness, reasonableness, model clarify capacity and responsibility at its centre. All the more explicitly, early examination concerning creating Network Intrusion Detection System utilizing an Explainable AI Frameworks (Mane & Rao, 2021) has been directed by Shraddha Mane and Dattaraj Rao.

As ML models offer improved accuracy, the intricacy increases and subsequently the interpret ability diminishes. In their paper, they fostered a profound neural network for, network interruption detection and proposed a reasonable AI system to exhibit model straightforwardness all through the AI pipeline. Using existing XAI calculations created from SHAP, LIME, Contrastive Explanations Method (CEM), that give clarifications on individual predictions, they applied the ways to deal with the NSL-KDD dataset showing effective expansion in model straightforwardness.

6. DATA-SET DESCRIPTION

UNSW-NB15 is an IoT-based network traffic dataset index with various classifications for ordinary activities and malicious activity practices from botnets (through characterization of attack type including Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode and Worms). The UNSW-NB 15 dataset were made by the IXIA Perfect Storm device in the Cyber Range Lab of the Australian Center for Cyber Security (ACCS) for producing a mixture of genuine present day typical activities and manufactured contemporary attack practices on IoT based organizations. Figure 1 shows the testbed design dataset and the strategy for the component production of the UNSW-NB15.

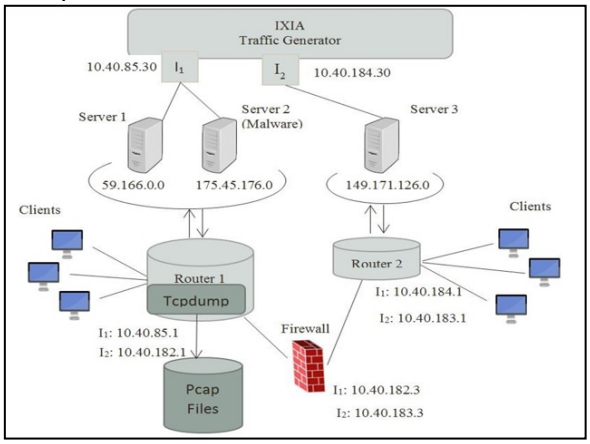
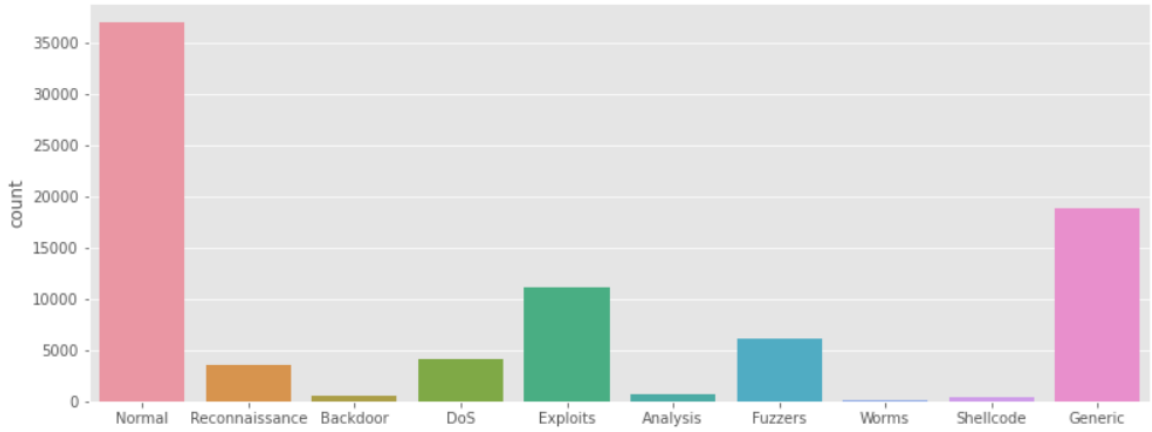


Figure 1. IXIA Traffic Generator Overview

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 behaviour for each record, attack, and normal behaviour. 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 behaviour. 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 behaviour. We could see that the dataset is adequately balanced for the binary response variable of activity behaviour.




```
The class ratio for the original data: 0.8:1 (37000/45332)
```

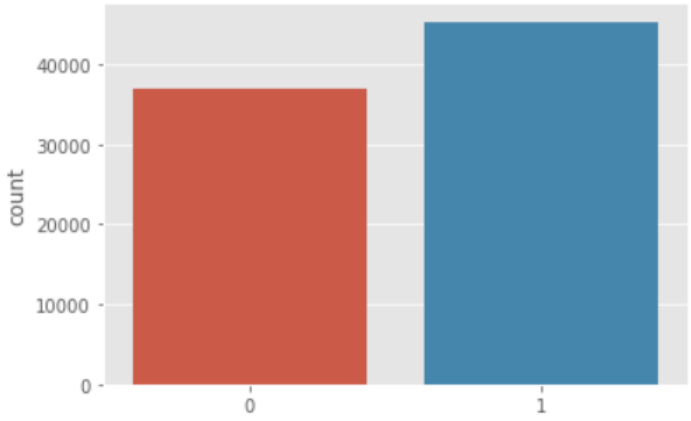


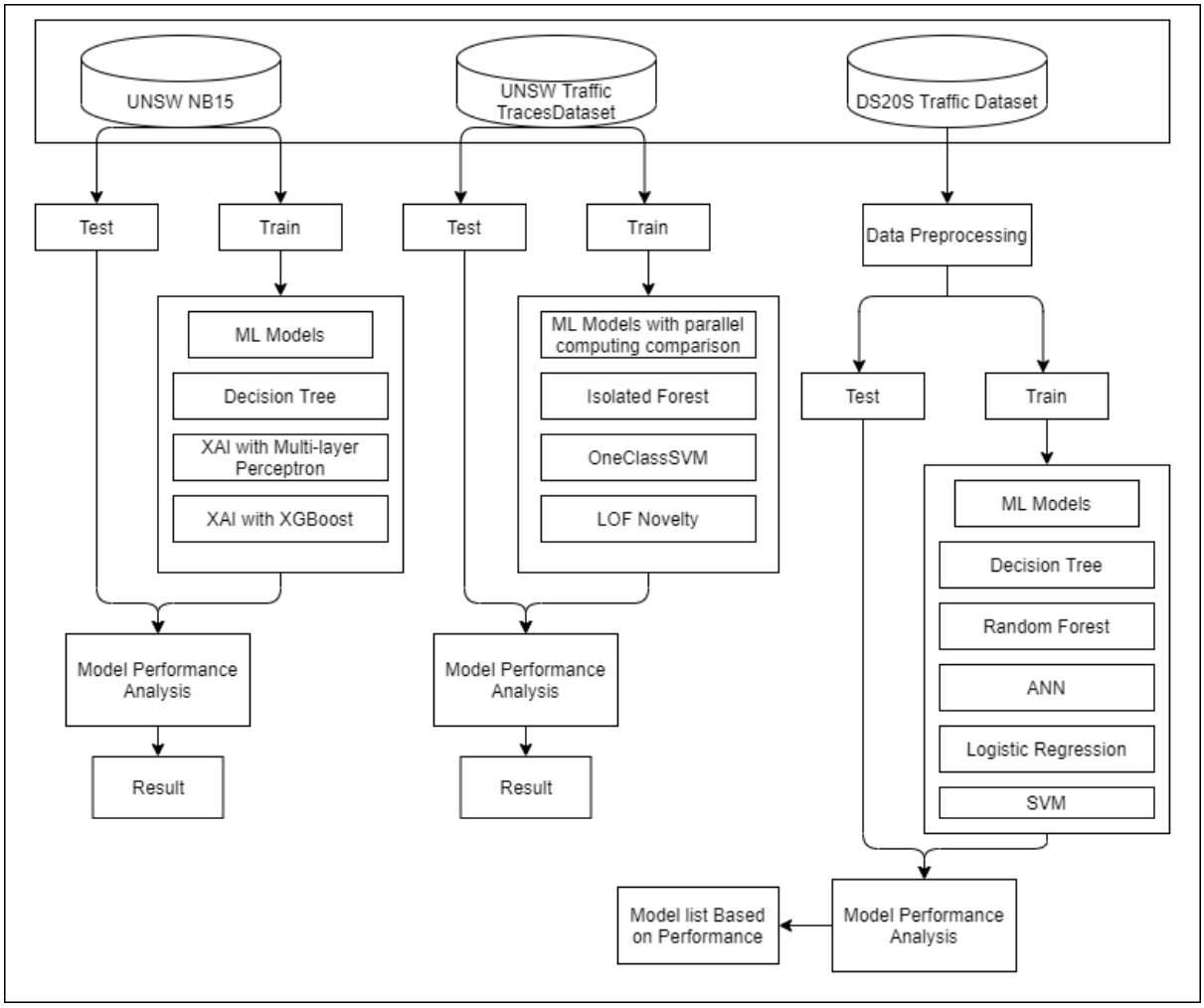
Figure 2. Training Dataset Distribution and Counts

7. SYSTEM ARCHITECTURE

The system consists of three modules:

1. Analysis on the UNSW NB15 dataset consisting of AI bots data using XAI
2. Analysis on UNSW traffic traces dataset using ML models with parallel computing
3. Analysis on DS2OS traffic dataset using ML models and ranking the models based on performance.

The system architecture of the proposed system is shown in the figure given below.



A) Analysis on the UNSW NB15 dataset consisting of AI bots data using XAI

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 behaviour 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 perceptron or single neuron models for complex predictive modelling 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 result of the network is contrasted with the expected results and an error is determined. This blunder is then proliferated back through the network, each layer in turn, and the weights are refreshed by the sum that they added 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 trade-off 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 application of gradient boosted decision trees intended 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 method consisting of new models being added to address the errors made by existing 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 modelling of Normal or Attack behaviour.

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) behaviour. 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 behaviour in the testing set. The following Python packages are and investigate to modify the ML classifiers for explain ability:

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.

B) Analysis on UNSW traffic traces dataset using ML models with parallel computing

Parallel and distributed computing are essential for the present day applications. In this module we execute the ML models in sequential and parallel computing, to be able to compare the outcomes. Applying parallel computing on these selected ML models will not only help understand the model, but also be able to compare till what extent can the execution time be reduced for a particular model.

We need to use numerous centres or various machines to accelerate applications or to run them at an enormous scope. Numerous instructional exercises disclose how to utilize Python's multiprocessing module. Sadly the multiprocessing module is seriously restricted in its capacity to deal with the prerequisites of present day applications. These necessities incorporate the accompanying:

- Running a similar code on more than one machine.
- Building microservices and entertainers that have state and can impart.
- Smoothly dealing with machine disappointments.
- Productively dealing with huge items and mathematical information.

Ray tends to these focuses, simplifies things straightforward, and makes complex conduct conceivable.

The ML models that will be computed parallelly:

i. Isolated Forest
ii. One Class SVM
iii. LOF Novelty

Other ML models that will be applied:

i. Random Forest
ii. Adaboost
iii. Gradient Boosting

C) Analysis on DS2OS traffic dataset using ML models and ranking the models based on performance.

For anomaly detection on IoT sensors on IoT sites, the following ML algorithms are applied:

i. Logistic Regression
ii. Decision Tree
iii. Random Forest

iv. ANN
v. SVM

The performance evaluation results with the list of ML models considering the execution time, accuracy, and other metrics.

8. RESULTS

A) Analysis on the UNSW NB15 dataset consisting of AI bots data using XAI

Decision Tree Classifier

The training set was used to build a Decision Tree classification model for Normal or Attack behaviour. The model performance accuracy against the testing set was 85% as indicated in Figure 3.

```
Accuracy: 0.8511072709748433
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
```

Figure 3: Decision Tree Classifier Report

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 determined as the reduction in node impurity weighted by the likelihood of arriving at that node. The most important features will be higher in the tree-like visualization generated.

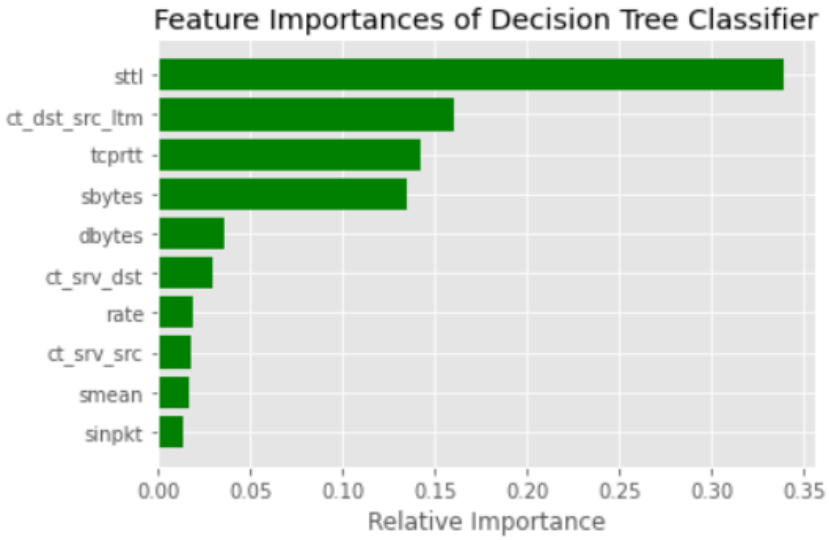


Figure 4. Decision Tree Feature Importance: Scikit Learn

Weight	Feature
0.2755 ± 0.0003	sttl
0.2417 ± 0.0009	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
0.0007 ± 0.0000	sinpkt
0 ± 0.0000	sjit
0 ± 0.0000	dinpkt
0 ± 0.0000	dttl
0 ± 0.0000	dloss
0 ± 0.0000	stcpb
0 ± 0.0000	sload
0 ± 0.0000	dload
0 ± 0.0000	rate
0 ± 0.0000	swin
0 ± 0.0000	dpkts
0 ± 0.0000	djit
0 ± 0.0000	is_sm_ips_ports
… *19 more* …	

Figure 5. Decision Tree Feature Importance: ELI5 Permutation Importance

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.

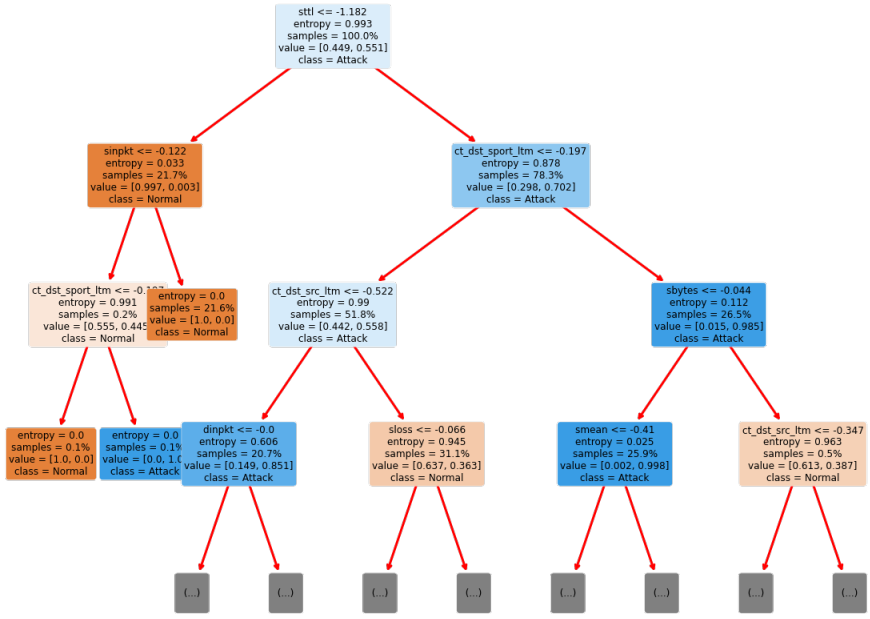


Figure 6. Decision Tree Classifier (Depth = 3 Nodes) Explainable AI Visualization

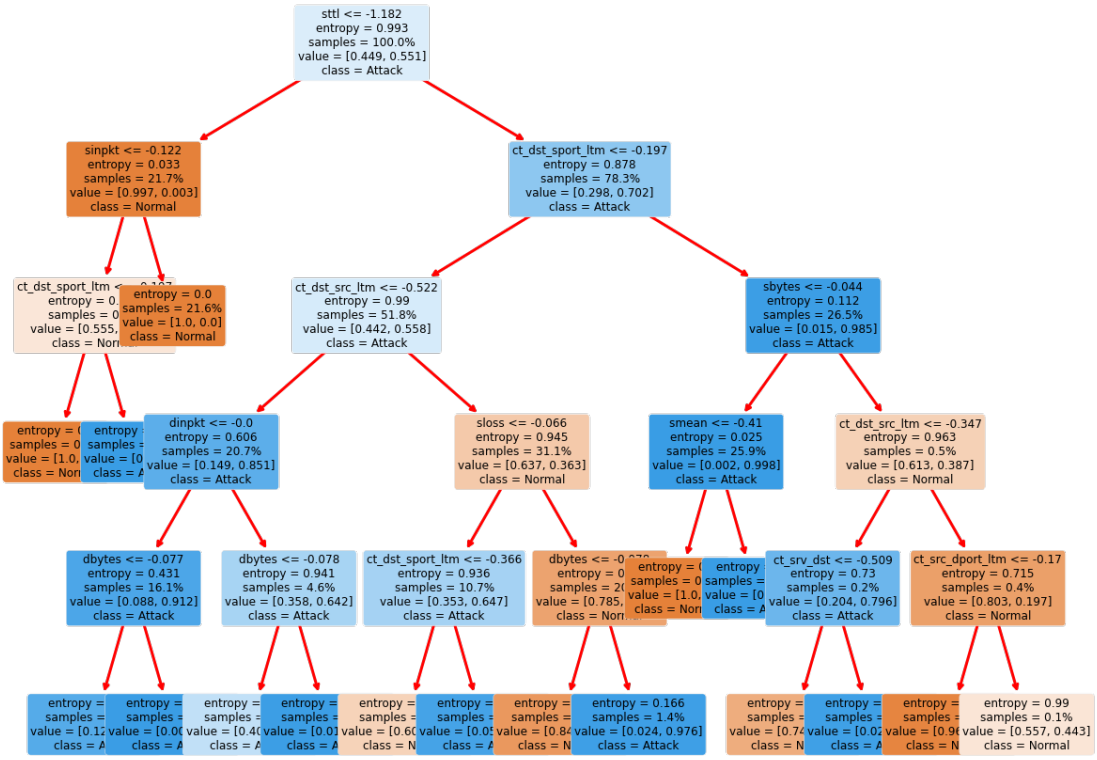


Figure 7. Decision Tree Classifier (Depth = 5 Nodes) Explainable AI Visualization

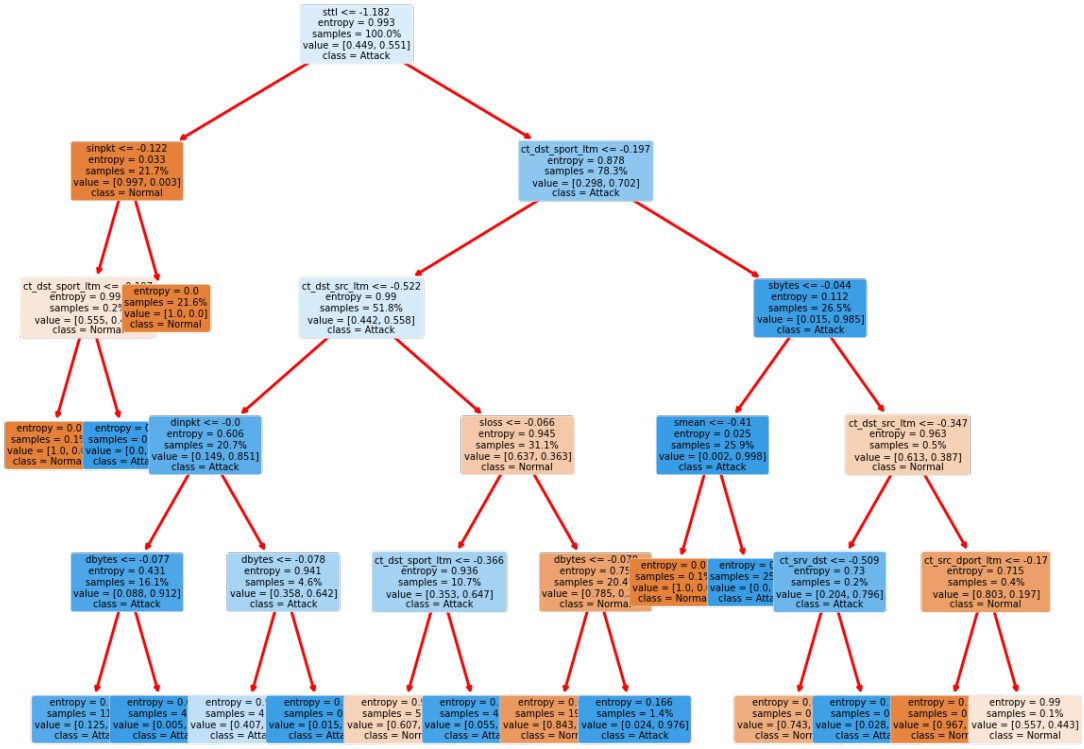


Figure 8. Decision Tree Classifier (Depth = 8 Nodes) Explainable AI Visualization

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 behaviour 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.

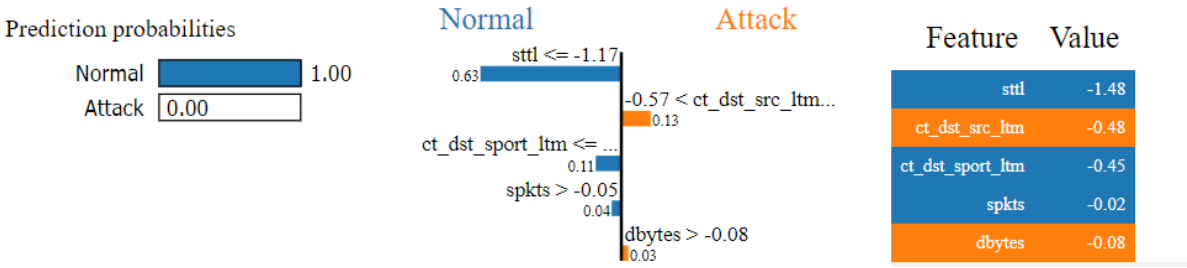


Figure 9. Single Classification Prediction using the MLP Classifier Explanation

The visual dashboard indicate which features and their weights brought the overall behaviour 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 behaviour. 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 behaviour, while closer to 0 indicates Normal Activity by a network traffic record.

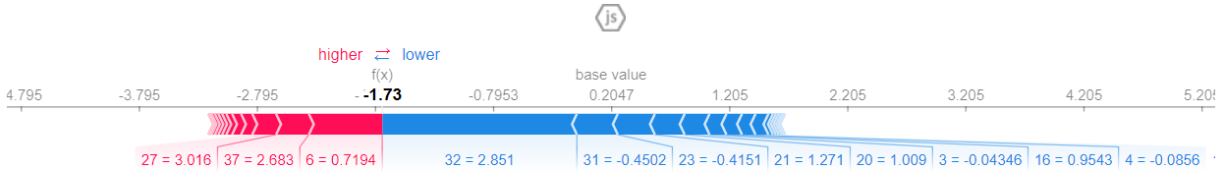


Figure 10. XGBoost SHAP- Visualize a single prediction

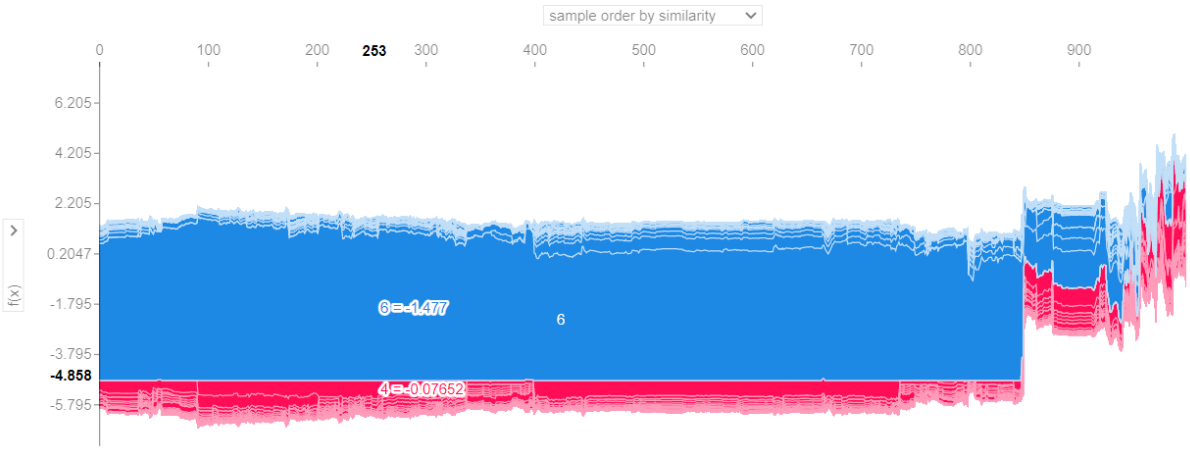


Figure 11. XGBoost SHAP - Visualize many predictions

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.

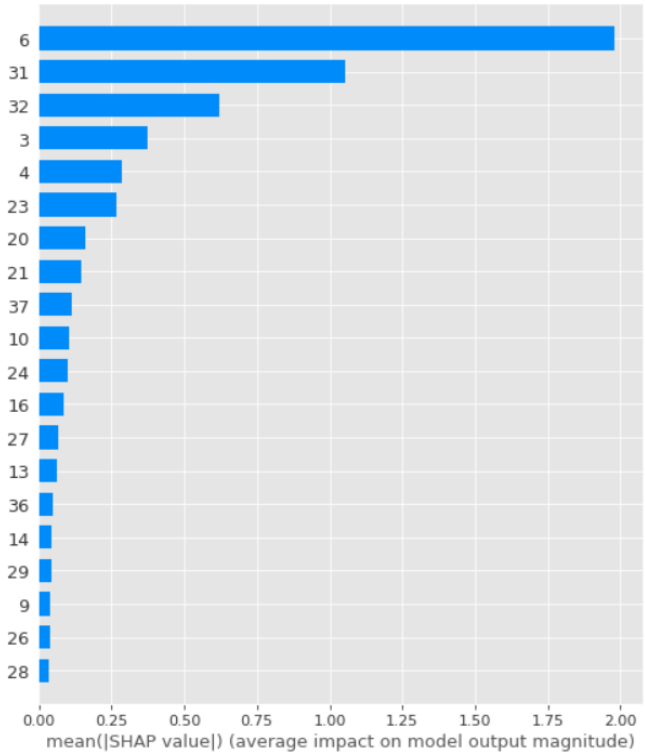


Figure 12. SHAP Feature Importance on XGBoost Classifier

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

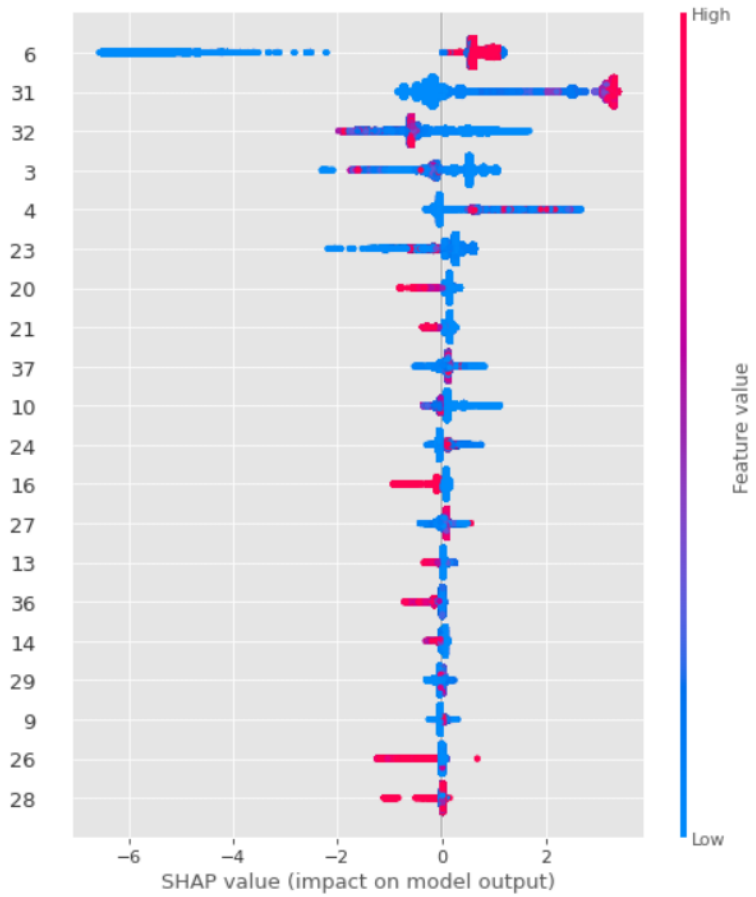


Figure 13. SHAP Summary Plot for XGBoost

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.

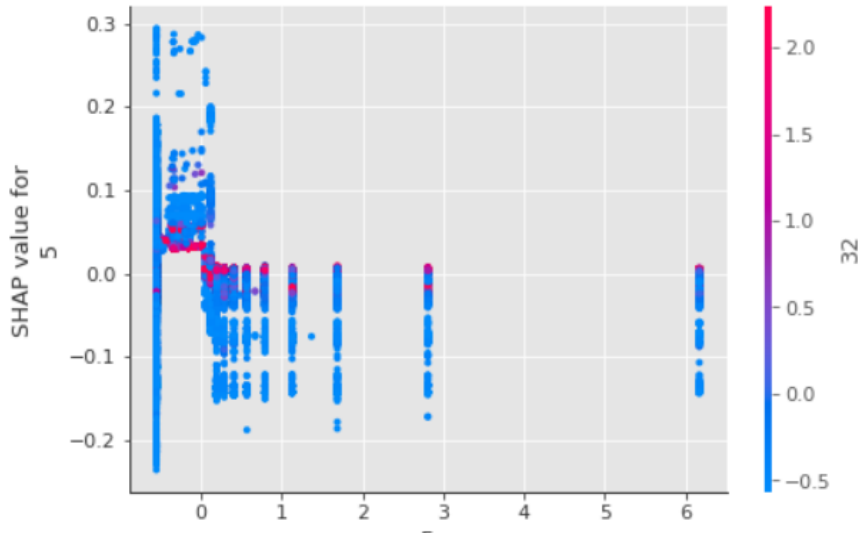


Figure 14. SHAP Dependence Plots for 'sttl' feature

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

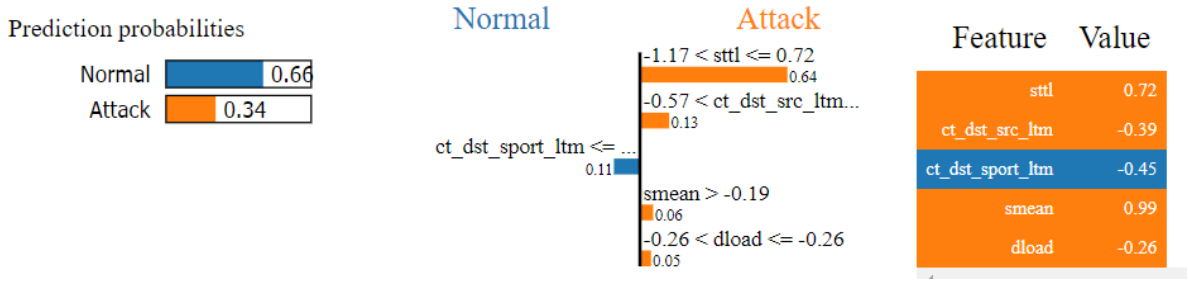


Figure 15. Single Classification Prediction using the XGBoost Classifier Explanation

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.

B) Analysis on UNSW traffic traces dataset using ML models with parallel computing

Sequential computing:

Isolated Forest	LOF Novelty	OneClassSVM

```
===== Iteration: 1 =====
### Metric Mix ###
TP:      99  22.50%
FN:       9   2.05%
FP:     259  58.86%
TN:      73  16.59%
Accuracy:    39.09%
Precision:  0.2765
Recall:     0.9167
F1 score:   0.4249

C:\Users\Admin\anaconda3\lib
nd will be removed in 0.24.
  warn(

===== Iteration: 2 =====
### Metric Mix ###
TP:      96  21.82%
FN:      12   2.73%
FP:     173  39.32%
TN:     159  36.14%
Accuracy:    57.95%
Precision:  0.3569
Recall:     0.8889
F1 score:   0.5093
===== Iteration: 3 =====
### Metric Mix ###
TP:      98  22.32%
FN:       9   2.05%
FP:     243  55.35%
TN:      89  20.27%
Accuracy:    42.60%
Precision:  0.2874
Recall:     0.9159
F1 score:   0.4375
```

```
===== Iteration: 1 =====
### Metric Mix ###
TP:      81  18.41%
FN:      27   6.14%
FP:      26   5.91%
TN:     306  69.55%
Accuracy:    87.95%
Precision:  0.7570
Recall:     0.7500
F1 score:   0.7535
===== Iteration: 2 =====
### Metric Mix ###
TP:      90  20.45%
FN:      18   4.09%
FP:      25   5.68%
TN:     307  69.77%
Accuracy:    90.23%
Precision:  0.7826
Recall:     0.8333
F1 score:   0.8072
===== Iteration: 3 =====
### Metric Mix ###
TP:      85  19.36%
FN:      22   5.01%
FP:      25   5.69%
TN:     307  69.93%
Accuracy:    89.29%
Precision:  0.7727
Recall:     0.7944
F1 score:   0.7834
```

```
===== Iteration: 1 =====
### Metric Mix ###
TP:       0   0.00%
FN:     108  24.55%
FP:       0   0.00%
TN:     332  75.45%
Accuracy:    75.45%
Precision:     nan
Recall:     0.0000
F1 score:      nan
===== Iteration: 2 =====
### Metric Mix ###
TP:       0   0.00%
FN:     108  24.55%
FP:       0   0.00%
TN:     332  75.45%
Accuracy:    75.45%
Precision:     nan
Recall:     0.0000
F1 score:      nan
===== Iteration: 3 =====
### Metric Mix ###
TP:       0   0.00%
FN:     107  24.37%
FP:       0   0.00%
TN:     332  75.63%
Accuracy:    75.63%
Precision:     nan
Recall:     0.0000
F1 score:      nan
```

Results compared of sequential and parallel computing:

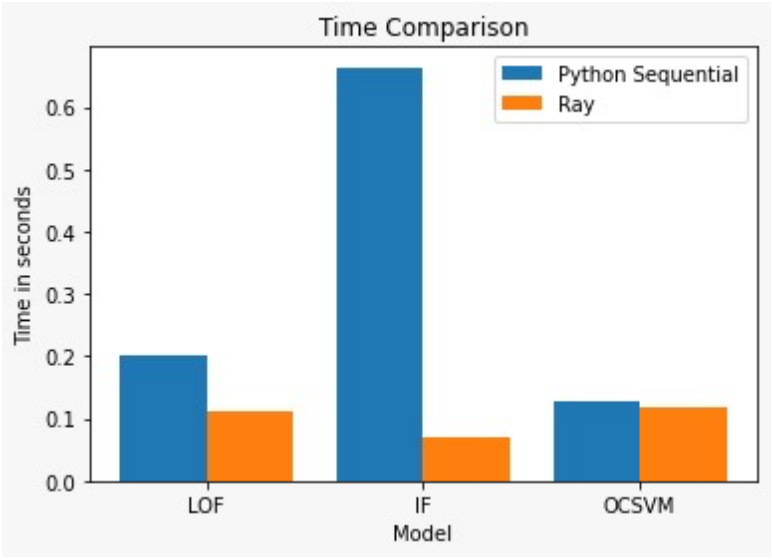
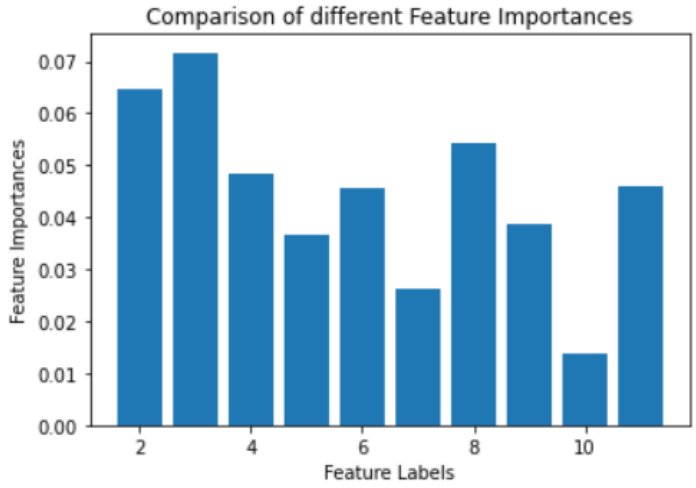


Figure 16. Time comparison chart

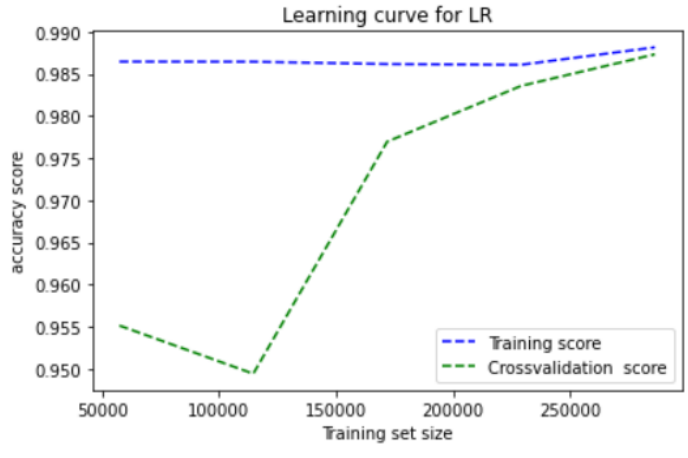
C) Analysis on DS2OS traffic dataset using ML models and ranking the models based on performance

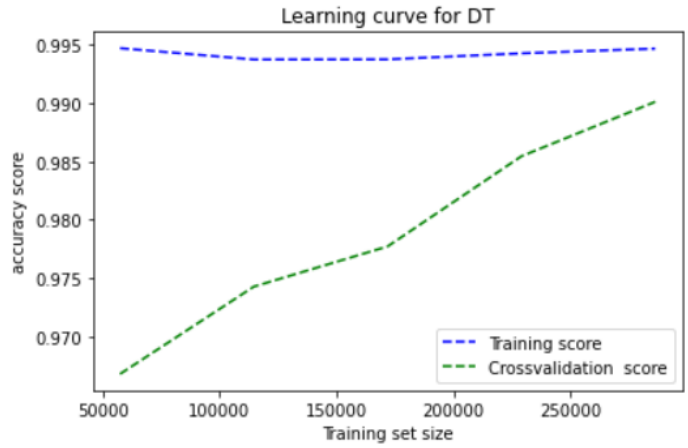
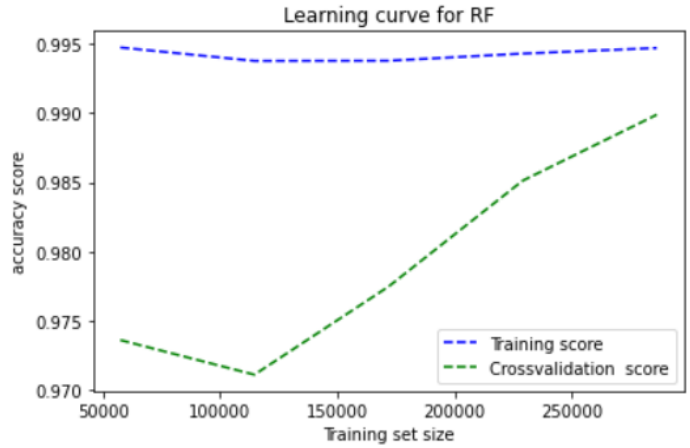
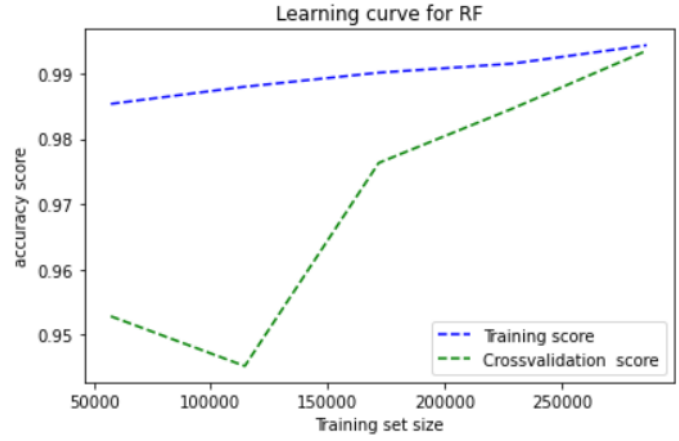
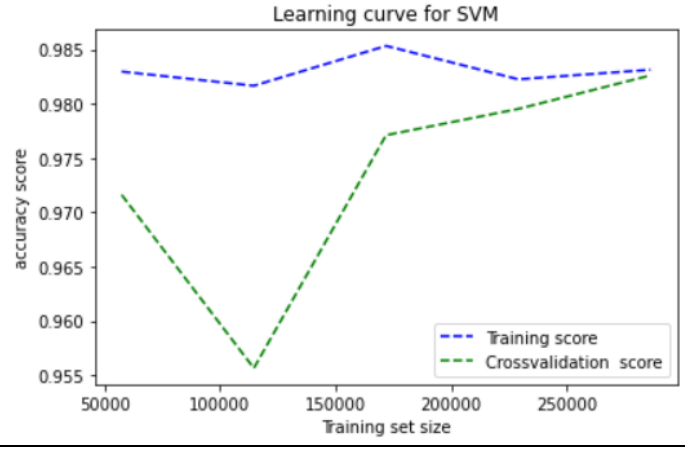



```
2  :  sourceAddress
3  :  sourceType
4  :  sourceLocation
5  :  destinationServiceAddress
6  :  destinationServiceType
7  :  destinationLocation
8  :  accessedNodeAddress
9  :  accessedNodeType
10  :  operation
11  :  value
```

Figure 17. Comparison of different feature importance

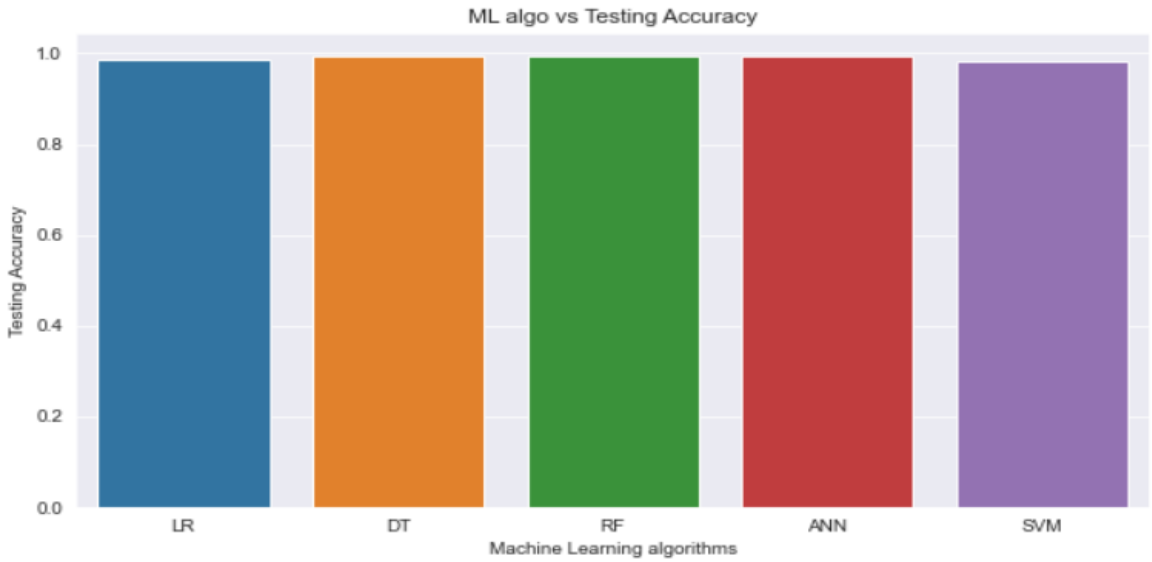
Learning curve for the models:

Logistic Regression	

Decision Tree	
Random Forest	
Artificial neural network (ANN)	
Support vector machine (SVM)	

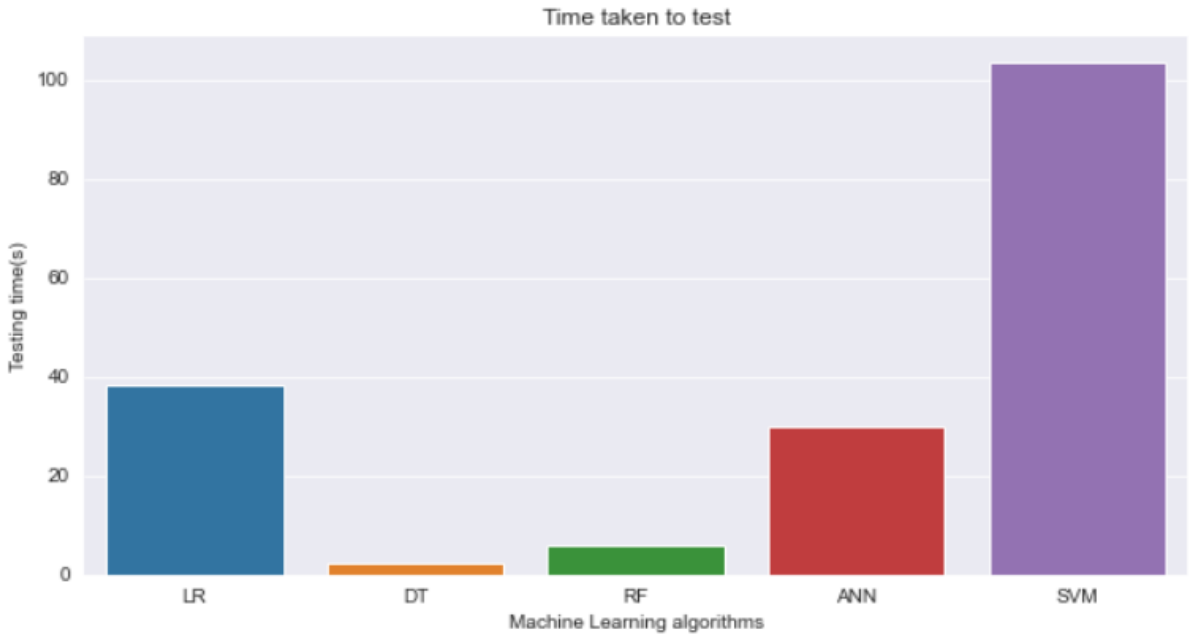
i. Comparing accuracy

Testing	LR	DT	RF	ANN	SVM
Accuracy	0.9874701420609312	0.9943566749081563	0.9943427062816913	0.9942588945229016	0.9827627149422398



ii. Comparing Time taken

Testing	LR	DT	RF	ANN	SVM
Time taken (s)	38.49999380111694	2.122478485107422	5.730687141418457	29.778282165527344	103.49215841293335



Based on testing accuracy most preferred algorithm (arranged from most to least accurate) :

Rank	ML algorithm
1	DT
2	RF
3	ANN
4	LR
5	SVM

Based on testing time most preferred algorithm (arranged from least to most time taken):

Rank	ML algorithm
1	DT
2	RF
3	ANN
4	LR
5	SVM

Thus it can be concluded that one should use DT technique on these kind of datasets for solving cyberattacks on IoT network because DT predicted D.P, M.C, M.O, SC, SP, W.S attacks accurately compared to other approaches.

9. 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 behaviour 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 impact the prediction of a cyber-attack and till what extent.

10. ACKNOWLEDGMENT

Words are in my most humble opinion, a token of gratitude and for this, I would like to say some. I would like to thank our faculty, Dr. Saira Banu J, for her providing with such a fruitful topic, and always guiding us. This project helped me gain insight into the subject of parallel and distributed computing. We got to know a lot about machine learning and its benefits. Apart from that, we would also like to thank our peers, who provided us constant support and kept us motivated throughout the project.

13. REFERENCES

Chkirbene, Z., Eltanbouly, S., Bashendy, M., Alnaimi, N., & Erbad, A. (2020). Hybrid Machine Learning for Network Anomaly Intrusion Detection. *2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies, ICIoT 2020*, 163–170. https://doi.org/10.1109/ICIoT48696.2020.9089575

Cristiani, A. L., Lieira, D. D., Meneguette, R. I., & Camargo, H. A. (2020, November 18). A Fuzzy Intrusion Detection System for Identifying Cyber-Attacks on IoT Networks. *Proceedings - 2020 IEEE Latin-American Conference on Communications, LATINCOM 2020*. https://doi.org/10.1109/LATINCOM50620.2020.9282320

Doshi, K., Yilmaz, Y., & Uludag, S. (2021). Timely Detection and Mitigation of Stealthy DDoS Attacks via IoT Networks. *IEEE Transactions on Dependable and Secure Computing*. https://doi.org/10.1109/TDSC.2021.3049942

Gu, T., Abhishek, A., Fu, H., Zhang, H., Basu, D., & Mohapatra, P. (2020). Towards Learning-automation IoT Attack Detection through Reinforcement Learning. *Proceedings - 21st IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, WoWMoM 2020*, 88–97. https://doi.org/10.1109/WoWMoM49955.2020.00029

Hodo, E., Bellekens, X., Hamilton, A., Dubouilh, P. L., Iorkyase, E., Tachtatzis, C., & Atkinson, R. (2017). Threat analysis of IoT networks Using Artificial Neural Network Intrusion Detection System. *ArXiv*.

Hwang, R. H., Peng, M. C., Huang, C. W., Lin, P. C., & Nguyen, V. L. (2020). An Unsupervised Deep Learning Model for Early Network Traffic Anomaly Detection. *IEEE Access*, *8*, 30387–30399. https://doi.org/10.1109/ACCESS.2020.2973023

Jatti, S. A. V., & Kishor Sontif, V. J. K. (2019). Intrusion detection systems. *International Journal of Recent Technology and Engineering*, *8*(2 Special Issue 11), 3976–3983. https://doi.org/10.35940/ijrte.B1540.0982S1119

Liu, F. T., Ting, K. M., & Zhou, Z. H. (2008). Isolation forest. *Proceedings - IEEE International Conference on Data Mining, ICDM*, 413–422. https://doi.org/10.1109/ICDM.2008.17

Liu, Z., Thapa, N., Shaver, A., Roy, K., Yuan, X., & Khorsandroo, S. (2020, August 1). Anomaly detection on lot network intrusion using machine learning. *2020 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems, IcABCD 2020 - Proceedings*. https://doi.org/10.1109/icABCD49160.2020.9183842

Mane, S., & Rao, D. (2021). *Explaining Network Intrusion Detection System Using Explainable AI Framework*. http://arxiv.org/abs/2103.07110

Maniriho, P., Niyigaba, E., Bizimana, Z., Twiringiyimana, V., Mahoro, L. J., & Ahmad, T. (2020). Anomaly-based Intrusion Detection Approach for IoT Networks Using Machine Learning. *CENIM 2020 - Proceeding: International Conference on Computer Engineering, Network, and Intelligent Multimedia 2020*, 303–308. https://doi.org/10.1109/CENIM51130.2020.9297958

Moussa, M. M., & Alazzawi, L. (2020). Cyber Attacks Detection based on Deep Learning for Cloud-Dew Computing in Automotive IoT Applications. *Proceedings - 2020 IEEE International Conference on Smart Cloud, SmartCloud 2020*, 55–61. https://doi.org/10.1109/SmartCloud49737.2020.00019

Sadreazami, H., Mohammadi, A., Asif, A., & Plataniotis, K. N. (2018). Distributed-Graph-Based Statistical Approach for Intrusion Detection in Cyber-Physical Systems. *IEEE Transactions on Signal and Information Processing over Networks*, *4*(1), 137–147. https://doi.org/10.1109/TSIPN.2017.2749976

Sahu, N. K., & Mukherjee, I. (2020). Machine Learning based anomaly detection for IoT Network: (Anomaly detection in IoT Network). *Proceedings of the 4th International Conference on Trends in Electronics and Informatics, ICOEI 2020*, 787–794. https://doi.org/10.1109/ICOEI48184.2020.9142921

Xie, M., Hu, J., Han, S., & Chen, H. H. (2013). Scalable hypergrid k-NN-based online anomaly detection in wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, *24*(8), 1661–1670. https://doi.org/10.1109/TPDS.2012.261

Yahyaoui, A., Abdellatif, T., Yangui, S., & Attia, R. (2021). READ-IoT: Reliable Event and Anomaly

Detection Framework for the Internet of Things. *IEEE Access*, *9*, 24168–24186. https://doi.org/10.1109/ACCESS.2021.3056149

Zidi, S., Moulahi, T., & Alaya, B. (2018). Fault detection in wireless sensor networks through SVM classifier. *IEEE Sensors Journal*, *18*(1), 340–347. https://doi.org/10.1109/JSEN.2017.2771226