Network Intrusion Detection: A Cooperative Security in The IoT Ecosystem Using Ensemble ML Algorithm

MSc Research Project
MSc Cybersecurity

Mariam Uleyele Isedu
Student ID: X22151079

School of Computing
National College of Ireland

Supervisor: Mr Joel Aleburu

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Mariam Uleyele Isedu

Student ID: X22151079

Programme: MSc in Cybersecurity **Year:** …2023 -2024

Module: MSc Research Project

Supervisor: Mr Joel Aleburu

Submission Due Date: 16-09-2024

Project Title: …Network Intrusion Detection: A Cooperative Security in the IoT Ecosystem Using Ensemble ML Algorithms

Word Count: 9511 **Page Count** 23

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: ……*MariamIsedu*……

Date: ……16th of September, 2024……

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Network Intrusion Detection: A Cooperative Security in The IoT Ecosystem Using Ensemble ML Algorithm

Mariam Uleyele Isedu
X22151079

Abstract

The rapid increase in the use of Internet of Things (IoT) devices across various sectors, industries and organizations has not only transformed the way we interact with the digital world, but it has also brought about notable security limitations. One of the major security drawbacks is the attack and spread of malwares like botnets that have the capabilities of been targeted towards DoS/DDoS attack, which can pose as a major threat to the confidentiality, integrity, availability and the entire security posture of any IoT ecosystems. This research paper presents a novel approach for DoS/DDoS attack detection in IoT ecosystem, employing a cooperative monitoring model designed in the form of a 'fog node' to prevent network intrusion. The proposed solution leverages a stacking ensemble machine learning technique, integrating the strengths of the Random Forest (RF) and Extreme Gradient Boost (XGBoost) algorithms as base models, and then subsequently restacked into a Random Forest model trained using the publicly available 'UNB CIC IoT 2023' dataset. The cooperative security nature of the model ensures node to node security monitoring of network traffic and enabling real-time attack identification in the form of anomalies on the network. Experimental results and evaluations demonstrate the effectiveness of our proposed model's high performance in terms of accuracy, recall, precision, F1-score and high detection rate with low false positive rate compared to single-layer models. This research contributes to the field of cybersecurity by providing a proactive, efficient and effective method for enhancing IoT network resilience through advanced machine learning techniques and cooperative defense mechanisms.

1 Introduction

The advancements in Internet-of-Things (IoT) technologies and gadgets are progressing rapidly, leading to a more convenient lifestyle by reducing and streamlining repetitive tasks. These IoT devices we use in our daily lives have the capacity to connect to one or more network and internet connectivity, they can collect data from their environment seamlessly, communicate and interact with each other within a shared network. The data collected by these devices are highly sensitive, they are analysed and exchanged within an IoT environment in real-time which poses a high risk and vulnerability for cyber-attackers.

Given the increasing number of organizations across various sectors like the financial, manufacturing, health, electricity, water and including both private and the public sector currently leveraging on IoT for application on building smart homes, smart cities, smart grids, smart banking and smart wearable trackers that have access to data like personal identifiable information (PII), health records, banking credentials and are able to share these

information via a network to various devices both within same network and outside it. According to the Gartner's report (2020), that from the year 2020 more than 80% of organizations have been utilizing IoT, which makes them vulnerable to IoT-related threats. And projection based on this report is that the number of IoT connections is expected to reach twenty-seven billion by 2025.

According to Ahmad et al. (2021), due to the interconnectivity that exist in an IoT environment it is prone to various security attackers ranging from physical attackers to software attackers, then network attackers and encryption attackers. These attackers may launch their attack via numerous means which include but not limited to man-in-the-middle (MITM) attacks, malicious node injection, RFID spoofing, phishing attack, banking Trojans, sinkhole attack, ransomware, poodle attack and cryptanalysis attacks. In addition, malwares like viruses, spywares, worms and remote access Trojans (RATs) that capable of infecting a device or network, turning it into a bot that can migrate into other nodes to form botnets and thereby leading to denial of service and distributed denial of services (DoS/DDoS) are not left out according to Chen et al. (2022).

DDoS attack is an attempt that is carried out with malicious intent to disrupt normal network traffic of a server or network by flooding it with excess traffic. Most DDoS attacks are carried out using botnets. The 2024 Global Threat Analysis Report executive summary published by Radware (2024), reveals that attackers are increasingly focused on conducting zero-day attacks, which can result in botnet-driven DDoS attacks. According to the report, the number of bot network intrusion attacks is on the rise at a rate of 12.9 million transactions per day. Additionally, the rate of DDoS attacks targeting consumers in the Europe, Middle East, and Africa (EMEA) area has grown to 43%, with an average of 106 incidents per month in the past year. This poses a greater risk if effective proactive network security measures, policies, standards, and technological innovations are not strategically put in place. According to a research carried out by Wazzan et al. (2021) on attack detection approaches, that there are three (3) phases of an IoT attack lifecycle: Scanning to locate vulnerable devices and networks, Propagation which involves bot installation and Attack which is the stage of executing malicious activities like DDoS. Also, from a research by Díaz (2020), botnets pose as high risk of increasing the capacity of DDoS attacks.

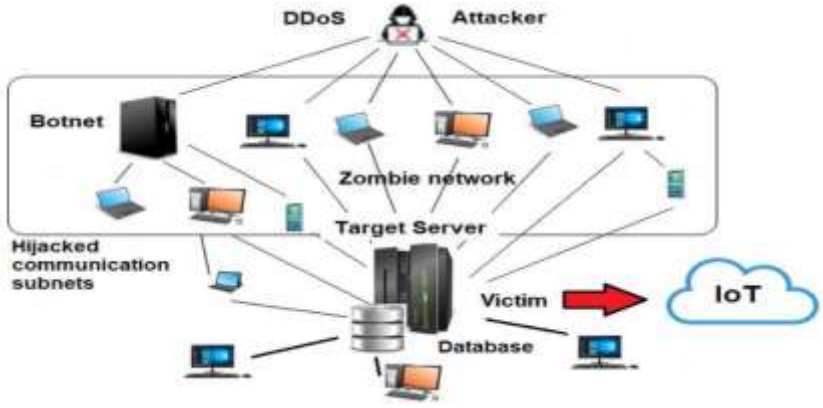

Figure 1: Flow diagram of an IoT Botnet-Driven DDoS Attack. **Source of diagram:** Díaz, (2020)

Figure 1 above demonstrates how cyberattacks launch DDoS attack on IoT devices and network through by first infecting a group of devices known as ‘botnets’ that are controlled by the attackers to carry out malicious activities.

Security of these IoT devices and their shared network between devices can be done through various means like the use of firewalls, enforcing customized security policies and privileges on network access control. But these traditional security measures are easily by-passed by attackers and are no longer sufficient as attackers keep coming up with sophisticated skills continuously. The use of blockchain techniques and machine learning models are two IoT security approaches currently been explored for security enhancement in the IoT ecosystem.

In the context of this research, the focus is on identification of ‘anomalies and signs’ in the network traffic of an IoT environment that are in the form of attacks and have the capabilities to exploit vulnerabilities and then form into attack launch that can lead to DDoS attacks. To achieve this, network intrusion detection techniques will be applied using machine learning (ML) to monitor the activities on the network traffic of an IoT environment to observe abnormal behaviours in the traffic. The ML model will be trained and tested with traffic patterns from IoT intrusion datasets that contains both malicious and benign instances. And the evaluation of the model will be carried out using various metrics and comparison to the current state-of-the-art as regard the subject matter.

Research Question

The above research problem motivates the following research question: How can cooperative security strategies and device network monitoring techniques, augmented by ensemble machine learning algorithms enhance network intrusion detection in IoT ecosystems and their network environments?

Research Objective

- To propose a very efficient model for network intrusion detection on IoT network ecosystem using a cooperative security approach in a form of a ‘fog node’ built using machine learning models.

The proposed resolution will explore the use of supervised machine learning classification algorithms in combination of ensemble ML techniques for network intrusion detection. Specifically, the Random Forest (RF), Logistic regression (LR), Support Vector Machine (SVM) and Extreme Gradient Boost (XGBoost) will be explored as the base models to be further passed on to a ‘Stacking’ ensemble ML technique leveraging the Random Forest algorithm for meta-learning on the 'UNB CIC IoT 2023 dataset’. All ML algorithms were chosen for the scope of this research because of their individual predictive strength, use of less resources and less and efficiency based on critical analysis of previous research. The implications of this research are substantial for the security of IoT networks. By deploying advanced machine learning models, we can achieve real-time DDoS attack detection, scalability improvement of intrusion detection systems and reduced incident response time.

This approach not only enhances security but also contributes to the overall efficiency and reliability of IoT networks.

The subsequent sections of this study is organized as follows: Section 2 is dedicated to the examination and assessment of relevant studies and related works pertaining to this project topic, conducted through a comprehensive literature review. Section 3 provides detailed information about the technique and methodology presented by this research and the stages of its implementation. Section 4 shows the design specification using the architectural diagram of the suggested model. Then Section 5 focusses on the discussion of the final implementation and outputs. Section 6 will focus on evaluation of the results from experiments using visualization tools and general discussion of significant findings. The concluding section of this report will be Section 7, it will provide a comprehensive summary of the research's significance, including its conclusions and suggestions for future studies.

2 Related Work

DoS and DDoS attack in an IoT network environment can be detected using various techniques, such as signature-based analysis like antivirus that employ the use of known malicious databases, rule-based systems that utilizes access control list (ACL) and firewalls, protocol analysis, network segmentation that involves isolating or segmenting the network into smaller segments in other to limit the spread of attack, the use of honeynets/honeypots-sinkholes in the form of devices deployed within the network to lure attackers in other to monitor their pattern, domain name server (DNS) monitoring, flow based and anomaly-based detection (Non-ML and ML-based) according to Pranav et al. (2024).

2.1 DDoS Attack Detection Techniques in the IoT Ecosystem

The use of honeypot and honeynet was employed to capture malicious requests within IoT network traffic in other to detect attacks like DDoS by Zhang et al. (2020) in their research. Three types of honeypots were proposed, first based on certain vulnerabilities, second is highly interactive and built from the IoT devices firmware that matches their vulnerabilities and the third is a multi-port honeypot designed from investigating the most exposed SOAP service ports in 2018. Although the proposed solution was simulated using IoT devices, it still has the limitations of automation and strengthening because of the ever-evolving patterns of attacks on the IoT environment. A signature-based traffic classification solution was introduced by Dimolianis et al. (2021) which has the capability to monitor, assess, and identify DDoS attack in a network environment. The solution employs a source-IP agnostic signature-based traffic filtering rules and classification can be then being done by machine learning models database with saved packet signatures for further analysis and classification. An important limitation of this approach is that the solution only focus on signature based DDoS attack features without taking into account the dynamic nature of attack source codes including the tactics, techniques and procedures used. Additionally, it was evaluated on the DNS traffic with attack only without considering other protocol attacks like UDP and ICMP.

Tandon et al. (2022) have introduced a scalable and less expensive open-source solution called AMON-SENSS for detecting DDoS attack that takes into account the network packets and flow characteristics. They utilized a hashed-based binning with multiple bin layer technique that is capable of carrying out traffic observation for traffic volume and notice change-points in the traffic flow. While this strategy yields effective results, its two main restriction is in the use of signatures like hashes which are subject to change and the regular signatures updates in other to meet with evolving attack mechanism.

2.2 DDoS Attack Detection in IoT Ecosystems using Machine Learning

Previous studies have been conducted to investigate and validate the efficiency of utilizing machine learning for detecting DoS/DDoS attacks in IoT devices and network environments. Ongoing research is also being conducted to further increase the overall security of IoT ecosystems with the use various machine learning approaches ranging from supervised ML approaches like classification and regression or unsupervised ML like clustering technique and deep learning approaches like the convolutional neural network (CNN) according to Ashraf and Elmedany (2021). Similarly, a comparative review of various ML algorithms for DDoS attack detection was carried by Chopra et al. (2021), the Random forest (RF), J48, Naives Bayes and ZeroR classifiers were explored on the Bot-IoT dataset and from the experiments, the RF classifier seems to outperform other models with an accuracy of 99.99% inaccurate detection. Although the research carried by Chopra et al. (2021) seems to pose the RF classifier is superior, there is need to explore more ML models, validate the chosen model performance with other datasets as these stands as drawbacks to this research.
Sujatha et al. (2022) conducted an experiment to identify DoS/DDoS attacks using various machine learning algorithms, including the Gaussian variant of Naïve Bayes, K-Nearest Neighbour (KNN), Random forest (RF) and XGBoost on the NF-UQ-NIDS-v2 dataset. The results showed that the RF algorithm achieved the highest accuracy with a score of 98.7%. However, from this research carried out by Sujatha et al. (2022), a significant limitation is that the RF algorithm was only compared to the other chosen models using their 'accuracy' on the same datasets, and calculating only accuracy is not enough to evaluate ML classifiers.

Also, Santhosh et al. (2023) proposed a model-based machine learning approach for identifying DDoS attacks. In their research, they utilized the RF model, XGBoost and introduced a modified version of XGBoost and assessed their effectiveness on the CICDDoS2019 dataset obtained from the Canadian Institute of Cybersecurity. Although the obtained findings suggest that the ML approach is effective, with the proposed classifier which is the modified XGBoost achieving an accuracy rate of 97% surpassing the other chosen models but the research did not account for the metrics used in carrying out the modification of the XGBoost classifier. In another research carried out by Srivastava et al. (2023), they proposed a framework for IoT based attacks like DDoS that explored the predictive power of the linear and non-linear Support Vector Machine (SVM) by considering the amount of data exchange between IoT nodes in a network, the transmission rate and lag in

message delivery. Based on the results from their research, the non-linear SVM performed better than the linear SVM but the draw back from this research is that the dataset and test-bed for models were not stated.

2.3 DDoS Attack Detection System (DDS) Using Ensemble ML Approach

Anomaly-based detection could be Non-ML or ML-based. Non-ML anomaly detection within an IoT network traffic is dependent on threshold or statistical based as there is already an acceptable baseline of behavior from the regular traffic which triggers an alert when exceeded. The use of ML-based network anomaly detection in IoT ecosystem is one of the most innovative recent means of IoT security as it makes it easy to detect unusual behaviour that could be signs of an attack according to Pranav et al. (2024).

A study was carried out by Dave et al. (2022), they proposed a fog computing DDoS detection framework for IoT environments to be introduced in the 'fog layer' and it utilizes both signature-based technique with a database from IP blacklist and anomaly-based technique that employs ML models like the decision tree (DT), Entra tree, XGBoost, RF and stacking ensemble ML trained and tested on the CICDDoS2019 dataset to classify normal or abnormal traffic flows. Although the proposed model which is expected to function as a fog node between the cloud nodes and IoT devices did achieve a good accuracy of 98.91% with XGBoost coming close at 98.90% but it was not neither tested against any benchmark dataset nor simulated in any other network. In another research carried out by Joseph Amalraj and Madhusankha (2023) using similar dataset as Dave et al. (2022), a hybrid model was that investigated the use of ensemble ML approach was introduced. In the research by Joseph Amalraj and Madhusankha (2023), the proposed model is expected to use 2 layers in its ensemble learning with the first layer been a combination of the DT, Logistic regression (LR) and KNN classifiers while the second layer which takes as input the blended results from the first layer uses only the RF model only. The results gotten from the research showed that proposed model did perform well with an accuracy of 99.9469% while the singular RF model was close with an accuracy of 99.9458%, but two major drawbacks from the research are the fact that only 21 features were selected from the dataset to train and test model, and the model utilized lots of resources been double layers.

Similarly, the attributes of the Random forest (RF) and XGBoost were explored for DDoS detection in IoT based networks in a more recent research by Srivastava et al. (2024). According to Srivastava et al. (2024) the approach seek to present a more robust approach that emphasizes on the advantage of ensemble ML techniques over the use of stand-alone models for attack detection. The model which integrated the predictive strength of RF and XGBoost, considered three main features which include the mean magnitude of data transfer for a node, transmission rate and duration of receiving message at the nodes in the network communications between a IoT devices with a network. Based on the result from their research, which was tested on a network traffic dataset gotten from IoT devices setup on MATLAB R2021a environment, their proposed was able to outperform other individual models with an accuracy of 99% using an optimized XGBoost. While this approach did

demonstrate interesting results but the limitation on feature selection, few IoT device setups and limited dataset used to test the model poses as drawbacks for the model generalization.

Although there seems to be commendable efforts through research already on-going with the use of ML and ensemble ML approaches for the detection of DoS/DDoS attacks in an IoT network. Other ML techniques considered but not utilized during our research include traditional ML algorithms like Decision Tree, Naives Bayes, clustering and deep learning techniques like the convolutional neural network (CNN) due to their constraints on the problem-type of this research, large amount of memory and computational resources. Previous research using these ML approaches all seem quite interesting. However, with DDoS been the target-attack from most launched cyber-attacks like spamming, phishing and other malware attacks, there is a great need to deal with the gap of having more efficient models that can detect the signs of DoS/DDoS attack in IoT ecosystem accurately without false alarms in from real-time network traffic.

Based on the scope of this research, we seek to explore the ‘stacking’ ensemble machine learning technique as a strategy for the suggested model. The stacking technique combines the predictions of two or more base models to get a final prediction. Stacking was chosen for this research because of its computational strength of been able to combine more than one ML algorithms. This approach is expected to be proactive, use less amount of memory space, less resources and sufficiently robust to detect the desired outcomes by lowering the risk of individual models and reduce false alarms of network intrusion detection systems (NIDS). The proposed model is expected to work as an ‘anomaly detector’ that can identify signs of DDoS attack within the IoT network traffic between devices (nodes) based on the volume of traffic on the same ecosystem. It seeks to optimize the existing models and the present state-of-the art approaches related to this research by boosting the confidence of individual model’s prediction decision.

3 Research Methodology

The research approach employed in this study is characterized as 'systematic' and is executed through a series of sequential steps where the results obtained from one stage are utilized as the input for the subsequent stages. Furthermore, processes can be iterated if necessary to achieve the optimal outcome while considering the project timeline. Creating a ML model can be achieved either with the use of supervised or unsupervised learning models. The major difference between the supervised and unsupervised is that - with the supervised ML approach there is labelled training data and baseline understanding of what the desired output should be unlike unsupervised that can be used to process un-labelled data which may take time to process and hence it’s possible to deliver inconsistent outputs. This research paper proposes a supervised learning model that utilizes ML classification models on an IoT dataset to make predictions on normal traffic or malicious traffic. The selection of this methodology was based on its simplicity in training the model, conducting analysis, easy identification of desired output, adaptation to new data and dynamic threats, easy result interpretation (benign

or malicious traffic) and enhanced decision-making processes for each phase. The following is the approach used to implement this model:

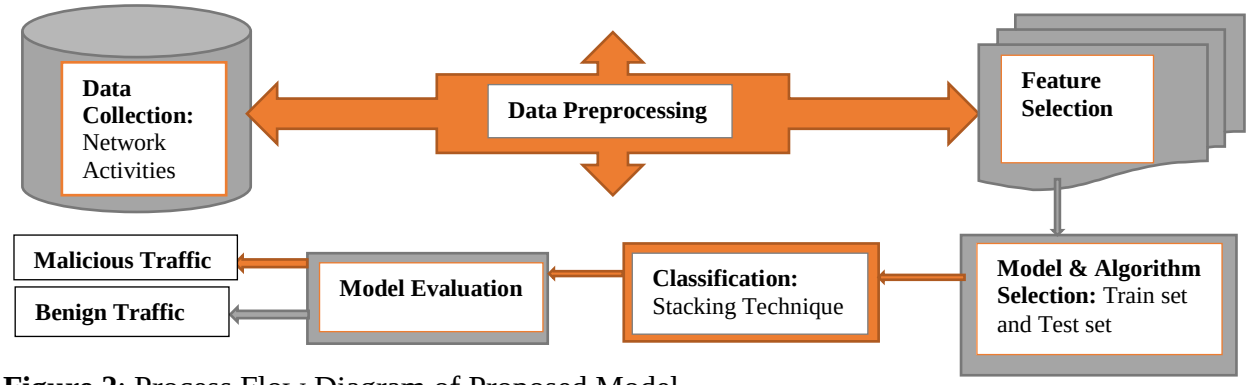

Figure 2: Process Flow Diagram of Proposed Model

Figure 2 above shows the systematic flow of how this research will be implemented from the point of data collection, through to all processing involved, evaluation and detection results.

3.1 Data Collection

An essential consideration in designing an efficient and robust machine learning model for DDoS attack detection is selecting an appropriate dataset and addressing the issues of creating a high-quality dataset that is well-suited for the proposed model. The selection of the dataset and the processing strategies have a significant impact on the outcomes of the final model. The dataset used in this study was obtained from the 'Kaggle' platform, an open source platform which is freely accessible to ML researchers. The 'UNB CIC IoT 2023 dataset' by Neto et al. (2023) from the University of Brunswick Centre of Cybersecurity was selected for this research after careful deliberation. This dataset was chosen over others because it more recent and has features of network traffic across 105 most used Internet of Things (IoT) devices with 7 types cyberattacks running on them including the Denial of service, Distributed Denial of Service (DDoS) which is our detection focus, botnets and brute-force attacks.

Some of the IoT devices considered from the ecosystem include but not limited to,

Category	**IoT Devices**
Home Automation	Netatmo Weather Station, LG Smart TV, Ring Alarm Keypad, Cocoon Smart HVAC Fan, AeoTec Doorbell 6, Govee Smart Humidifier
Hub	SmartThings Hub, Ring Base Station, Philips Hue Bridge
Audio	Amazon Alexa Echo Dot, Google Nest Mini Speaker, Sonos One Speaker
Camera	Nest Indoor Camera, TP-Link Tapo Camera, Eufy Doorbell Camera
Sensors	Fibaro Motion Sensor, Fibaro Flood Sensor, AeoTec Water Sensor
Lightings	Lumiman bulb, Teckin Light Strip, Philips Hue White
Power Outlet	GoSund Smart plug, Amazon Plug, Yutron Plug, Teckin Plug
NextGen	Raspberry Pi 4

Table 1: Categories of IoT Devices Considered

Dataset: The dataset for this research is already anonymized in other to protect identity and sensitive other of device users. It will be divided into two subsets, one for training and the other for testing. The features of the dataset have key attributes that covers the scope of this research. Some of the valid variables include;

1) ts – represents the timestamp of first packet in flow
2) Rate – shows the rate of packet transmission in a flow
3) Min, Max, Avg and Std – represents the minimum, maximum, average and standard deviation lengths in the flow
5) Flow_duration – the time between first and last packet received in flow
6) Rst_count – shows the number of packets with rst flag set in the same flow
7) SSH – this indicates if the application layer protocol is SSH
8) Protocol_type – Protocol numbers, as defined by the IANA. Ex: 1 = ICMP, 6 = TCP
9) ICMP – this indicates if the network layer protocol is ICMP
10) Number – this is the number of packets in the flow
11) Covariance – the covariance of the lengths of outgoing and incoming packets
12) Weights – the number of incoming and outgoing packets
13) Label – this identifies the '**type of attack'** or '**Benign traffic**' which means no attack

3.2 Data Pre-processing

Strategic data pre-processing is a crucial step that must be considered while preparing datasets for machine learning models. To achieve this task efficiently, it is necessary to utilize pre-processing strategies like oversampling or under-sampling depending on the granularity in the datasets. Methods such as attribute scaling to eliminate unnecessary attributes and outlier removal, addressing missing values can be explored. For this research the Synthetic Minority Oversampling Technique (SMOTE) will also be used based on the dataset size, data types, and dimensions.

In other to transform and prepare the dataset for this research, the following steps will be considered:

Data Cleaning: exploration of the dataset, datatypes and checking for missing values or nulls to be handled.

Data Categorization and Sampling: this involves identification of the labels, scaling, standardization, sampling, oversampling (SMOTE) to balance the classes and encoding categorical data into numbers as required.

Data Encoding: It is necessary to translate the data into numerical format for easy computation. The class label consists of distinct categorical values, including 7 attack-classes and one benign class. Prior to executing the algorithms, these values will be encoded into numerical values. This research employed the Label encoder approach to encode the categorical values in the class label.

3.3 Feature Selection

Selecting the necessary features best for the research scope is very essential as it has major implications on the results of all experiments. Before carrying out feature selection it is of

best practice to first get a detailed statistical description of the dataset that shows how each variable is dependent on the output variable.

Correlation-Based: In this research, the correlation-based feature selection (CFS) technique will be employed. This technique generates a correlation that reveals the intricate relationship between variables and the data. It was chosen because its ability to identify features that have a strong correlation with the goal variable (output) and its capability to choose the most relevant features to attain the required outcomes. The correlation-based technique was chosen as against the other features selection approaches explored, like the variance threshold technique which removes features from a dataset by assumption and this can lead to removal of importance features that may be useful to some ML-algorithms, Chi-squared test technique which functions by evaluating dependencies of features and Analysis of variance (ANOVA) technique which functions by comparing mean values of various groups of features, hence works best for dataset with majorly numerical features.

3.4 Classification Model and Algorithms Selection

During this research, all ML algorithms and ensemble learning technique were chosen after critical literature review of previous research works. Although there are several ensemble techniques like boosting, voting, bagging and stacking. The ensemble technique chosen for this research is 'Stacking or Meta-learning'. This technique was chosen because it aggregates the outcomes of each individual prediction made by the base algorithms (RF and XGBoost) and then combine the result into another ML algorithm which for the purpose of this research is the Random Forest (RF) algorithm. The stacking technique can assist reduce the limitations of each base algorithms and utilize their complimentary capabilities to improve their DDoS attack detection rate.

Stacking is effective in reducing false positives by training the meta-learner to identify the superior performance of each model under varying instances. Random Forest may generate false positives when there is a significant amount of variability in the data. However, XGBoost can address this issue by placing greater emphasis on rectifying these faults. Our stacking model leverages the strengths of both models to minimize the occurrence of false alarms.

Also, each model exhibits distinct error detection patterns, so the stacking can combine their predictions to generate a more precise final prediction. Random Forest and XGBoost capture distinct patterns and features in the dataset, enabling them to identify both simpler and more intricate attack methods when used in combination

All base algorithms chosen are based on their individual computational and classification strengths:

Support Vector Machine (SVM) – The SVM is a supervised ML algorithm that is widely recognized for its capacity to handle both linear and non-linear correlations. According to (Nguyen et al., 2020), its main technique is in identifying the best hyperplane that best

separates the classes. It was chosen for use on this research because of its capacity to accurately identify intricate patterns in the data and improve the accuracy of classification outcomes.

Extreme Gradient Boost (XGBoost) – the XGBoost employs the boost ensemble ML technique and so has the capacity to enhances the performance of weak learners like random forest and other boosting algorithms according to (Singh et al., 2024). Also, it effectively handles categorical data structures, missing values in highly dense dataset by eliminating the need for complex data transformation as outlined by Singh et al. (2024). Additionally, in a research by (Santhosh et al., 2023), the has been identified as been able to build the best models when combined with weak models.

Random Forest (RF) – the RF machine learning algorithm is obtained from the combination of multiple decision trees that are constructed using a randomly selected sample of data. According to (Alghamdi and Bellaiche, 2022), predictions are generated from each individual tree and the best answer is determined by a voting process. It was chosen because it has in-built metrics to identify feature importance and has good model interpretability.

Logistic Regression (LR) – According to Mousavi et al. (2023), this ML algorithm possesses a framework that is probabilistic and exhibits both great computing efficiency and scalability, hence aids easy implementation and interpretation of model. It maps the predicted values to the probability values ranging between 0 and 1, hence it excels at handling both binary and multiclass classification problems.

3.5 Model Training and Testing

The next phase after the feature selection, is to split the dataset into two parts: which are the '**train**' set and the '**test**' set. On the basis of this research, the standard 80/20 percentage was considered. 80% of the dataset was allocated for training, while the remaining 20% was allocated for testing. The function *'train_test_split'* from the Scikit learn package was utilized, as shown in the figure above. The selected models, as previously stated based on the research scope, were implemented on all sub-dataset accordingly. Each base algorithms and proposed model were evaluated for training using the *'train sub-dataset'* and predictions were experimented on using the *'test sub-dataset'*.

3.6 Model Evaluation

The evaluation metrics selected in this research were chosen because they are best for binary classification problems. The prediction outcome expected from each experiment carried out during this research is either ‘attack type = DDoS’ or ‘benign traffic or any attack’. All base models and the proposed model will be evaluated using several metrics and a final simulation of the proposed model using a network simulation tool. According to Gupta (2023), the performance of an ML models can be evaluated using these six (5) metrics;

Confusion Matrix: the confusion matrix is like a table that comprises of the true positives, false positives, true negatives and false negatives predictions by a model. Whereby;

True positive (TP) – values predicted by a model that are actually true and predicted correctly.
False positive (FP) – values predicted by a model as true but are actually not true or not correct.
True negative (TN) – when a model predicts values as 'not true' and its actually 'not true'
False negative (FN) – when 'true values' are predicted as 'not true' by a model

Accuracy: Accuracy is a metric that quantifies the ratio of accurately classified data by a model to the total number of data instances. The formula for calculating accuracy is:
Accuracy = (TN + TP) / (TN + FP + TP + FN)

Precision: the precision metrics is used to measure the number of truly correct predictions by a model among the total number of correct predictions. A precision value of 1 indicates that the false positive rate is very low, approaching zero. Also, precision highlights a model capability to correctly identify TPs without misclassification of negatives. The formula for calculating accuracy is: Precision = TP / (TP + FP).

Recall: Recall can also be referred to as sensitivity, is the ratio of accurate positive predictions by a model to the total number of positive samples in the dataset. A recall result of 1 should be the objective of every effective model, as it indicates that the number of false negatives (FN) is extremely insignificant or possible null. The formula for calculating accuracy is:
Recall = TP / (TP + FN).

F1-Score: The F1-score is calculated as the equal-weighted average of Precision and Recall. Its purpose is to optimize the trade-off between precision and recall. For it to reach a value of 1, both precision and recall must be equal to 1. The formula for calculating accuracy is: F-1 Score = 2 * ((precision * recall) / (precision + recall)).

Simulation and Validation: In addition to the above evaluation metrics, the proposed model will be simulated using network simulation tools. During this research, the Mininet network simulator will be employed to analyse the general performance of the final model in other to check for the 'network intrusion detection accuracy and frequency rate' of detection. After several consideration of other tools like the OMNeT++, Scapy, EmuEdge and NS-3, the 'Mininet' was chosen for this study because of its graphical user interface (GUI) process flow, easy integration with objected-oriented programming languages. According to Winz et al. (2023), the Mininet simulator is one of the best that have recently gained traction form researchers as it is able to demonstrate realistic network topology for network security simulation, its open-source, uses less computer resources and allows for custom simulations which includes DDoS and other cyber-attacks.

4 Design Specification

The proposed model architectural diagram illustrates the processes and workflow of the several phases involved in the whole development and simulation of the model.

In other to achieve the aim of this research, which is to successful identify any behaviour that can lead to DDoS attack within an IoT network traffic. The proposed model is expected to act as anomaly-based intrusion detection '***Fog node'*** and intermediary between IoT devices in the same ecosystem that can detect any sign of DoS/DDoS attack. The fog node can either be implemented in way whereby it sits between the 'Cloud Service/Public Internet' and the connected devices or implemented as multiple fog nodes between the network traffic of communication from one connected device to the other in the ecosystem. This is to allow for a proper cooperative security between devices.

The fog node in this concept, collects the data from the cloud/internet or from the network activities between devices and processes it within the shortest possible time using the proposed final model. Based on the results from it processing outcome, it should trigger an alarm if an intrusion attack that has any signs of DoS/DDoS attack is detected in the network traffic and possibly be able to notify the traffic volume rate and frequency, identify any other form of attack that can be related to the DDoS or when there is no trigger then it's a benign traffic.

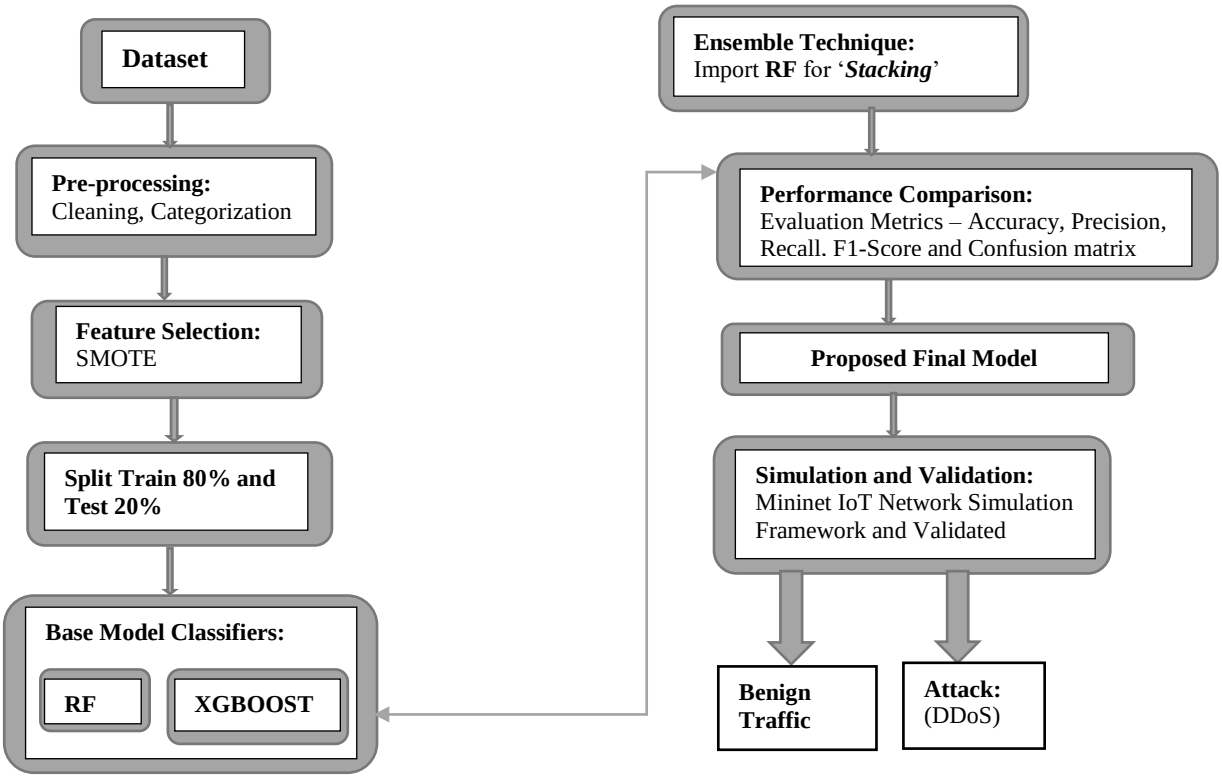

Figure 3: Proposed Model Design Specification

Based on the Figure 3 above, the fog node can be introduced into the IoT ecosystem as a node that utilizes our proposed model, that is the Stacking (RF+XGBoost=> RF).

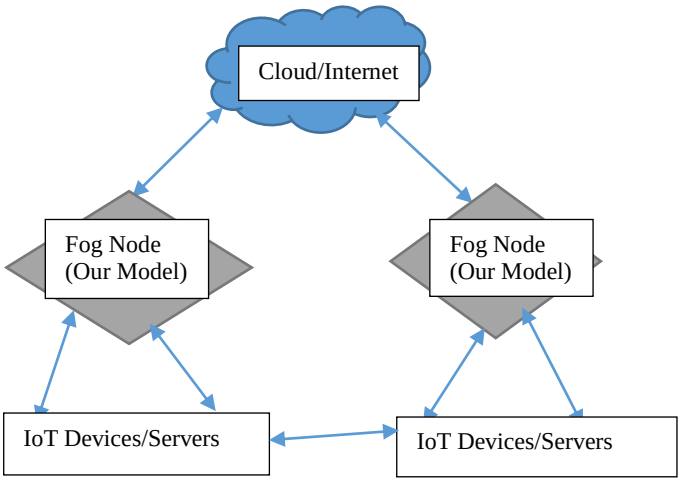

Figure 4: Fog Node Integration

From Figure 4 above, the fog node should be able to process network traffic from the internet or traffic flow between two or more nodes communication in the fastest possible time using our proposed stacking model as its monitoring technique to identify and detect any form of anomalies that appear to have signs of DDoS attack, and then trigger an alarm based on the traffic frequency rate.

5 Implementation

In this section, we practically demonstrate the methodology, and steps carried out using various platforms to achieve our proposed model.

5.1 Prerequisites and Basic Requirements

Requirements: The open-source Google Colab (GC) platform on a Dell PC running Microsoft Windows 10 Pro was used as the development and learning environment for our research and the ML model creation as against the Anaconda Navigator and Jupyter Notebook because of processing time efficiency and auto-save features of the GC platform. Other requirements include:

- Programming language - Python-3
- PC Processor - Intel Core i7
- System Memory - 16GB

Python Packages Installed: Below are some of the libraries and packages installed to aid the implementation of this research and easy achievement of required results:

- **Pandas** - Pandas is a Python library designed for the purpose of analysing data. The package provides a wide range of methods for manipulating numerical data.

- **NumPy** - NumPy, short for Numerical Python, is a Python library that is utilized for the purpose of exploring arrays of data. With NumPy, arrays manipulation can be achieved quicker compared to the use of ordinary Python lists.

- **Scikit-learn** – Usually imported using the form 'sklearn', is one of the most valuable Python libraries for machine learning. The sklearn toolbox comprises a variety of very efficient statistical and ML modelling techniques which can be used for classification, regression, clustering and dimensionality reduction problems.

- **Matplotlib** - This toolbox from Python provides easy interpretable and interactive visualizations. It also enables the solution of both complex and simple dataset exploration issues.

- **Seaborn** - The Seaborn library is built around a foundation that enables the creation of graphical plots. It will be utilised for observing stochastic distributions.

- The '***tqdm***' Library – With the 'tqdm' library, iterative processes and long-running task can be tracked.

5.2 Dataset Description

The dataset explored for this research was collected from the network activities of various IoT devices as described above in Section 3.1. As observed, the dataset contains various attacks including 'DDoS', 'DoS and botnets among others hence making it suitable for the scope of this research. The entire dataset CSV folder contained 169 parts of datasets in excel sheets with the same number of labels (47 columns) but different instances. The 'Part 00000' with 238688 rows was chosen for this research because of easy iteration, faster processing time, quick balancing and analysis, simplified debugging and classification.

Dataset Cleaning: The cleaning of dataset was done using various actions like the 'drop duplicates' to ensure no repetition of instances, then we checked for missing values using '*isnull and non-null counts'*. Additionally, standardization and scaling were carried out to ensure equal contribution from features, and label encoding was also done to convert categorical labels to numerical format in other to simplify the dataset, fast computation, easy interpretability and memory efficiency.

Label	Value Count		Label	Value Count
DDoS-ICMP_Flood	36554		DDoS-UDP_Fragmentation	1484
DDoS-UDP_Flood	27626		DNS_Spoofing	925
DDoS-TCP_Flood	23149		Recon-HostDiscovery	697
DDoS-PSHACK_Flood	21210		Recon-OSScan	517
DDoS-SYN_Flood	20739		Recon-PortScan	430
DDoS-RSTFINFlood	20669		DoS-HTTP_Flood	414
DDoS-SynonymousIP_Flood	18189		VulnerabilityScan	210
DoS-UDP_Flood	16957		DDoS-HTTP_Flood	169
DoS-TCP_Flood	13630		DDoS-SlowLoris	106
DoS-SYN_Flood	10275		DictionaryBruteForce	63
BenignTraffic	5600		SqlInjection	31

Mirai-greeth_flood	5016		BrowserHijacking	30
Mirai-udpplain	4661		CommandInjection	28
Mirai-greip_flood	3758		Backdoor_Malware	22
DDoS-ICMP_Fragmentation	2377		XSS	18
MITM-ArpSpoofing	1614		Uploading_Attack	8
DDoS-ACK_Fragmentation	1505		Recon-PingSweep	6

Table 2: Dataset labels and Value counts

From Table 2 above, the dataset appeared to have similar labels like the DDoS attacks, Mirai botnet, spoofing, Recon and Web attacks. Hence the need to remap them into relatable labels of malicious attacks, that is DDoS, DoS, Mirai, Spoofing, Recon, Web, Brute-force and Benign for easy processing.

5.3 Dataset Pre-processing

During the dataset analysis we noticed that it was imbalanced, hence the need to carry out re-sampling. During this research, the oversampling technique -SMOTE was used. SMOTE technique was used because it generates synthetic data to make up the required chosen samples. In the implementation of our model, we first ensured the '**imblearn**' library was installed and upgraded using the '*pip install scikit-learn imbalanced-learn*' command.

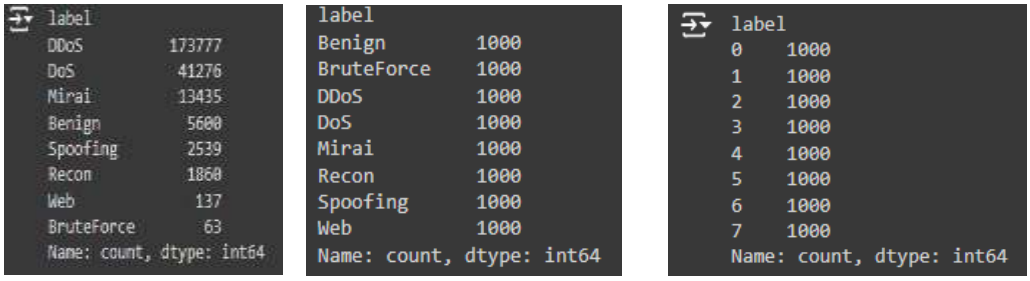

Figure 5: Before SMOTE application (5a), After SMOTE application (5b) and after Label encoding (5c)

From the Figure 5 above, after the SMOTE application we had 1000 samples each from each attack-type class, we encoded the categorical values in the label class to make it easier to compute the ML models using 'Label Encoder' technique. This resulted to each attack category remapped to numerical value ranging from 0 to 7, with **Benign** denoted as '**0**' and '**DDoS'** noted as the '**number 2'** attack on the row.

5.4 Feature Selection

Choosing significant features for training our model is a highly important process in machine learning classification. This research used CFS feature selection to enhance precision and reduce training times by training our model with the optimal number of relevant features and disregarding irrelevant ones. Based on the concept of CFS, features that have a significant correlation with the target variable are more suitable for producing accurate predictions compared to features that have a poor connection.

Analysis of Feature Importance: The feature importance and priority given to features for prediction is dependent on each ML algorithm. Features that were most considered for prediction for one model e.g, the XGBoost, may be the least considered in another ML model like the RF. See below the top 5 features consideration by each model employed for this research:

Random Forest	Logistic Regression	XGBoost	SVM
IAT	Flow_duration	Magnitude	Flow_duration
Magnitude	Rst_count	SSH	Rst_count
Header_lenght	Header_lenght	IAT	Magnitude
Rst_count	Variance	Number	Protocol type
Flow_duration	Ack_flag_number	ICMP	Covariance

Table 3: Top 5 Features Considered Across all Base Models

From Table 3 above, we can see that 5 features with major influence across the individual models include but not limited to:

Flow_duration – time between first and last packet received in flow
Rst_count – number of packets with rst flag set in the same flow
Magnitude – Sqrt (Avg of the length of incoming packets in the flow + Avg of the length of outgoing packets in the flow
IAT – the time difference with previous packets
Header_lenght – length of packet headers in bits

5.5 Model Training, Testing and Stacking

Training and Testing: Following the feature selection phase, the next step is to divide the dataset into a 'train' set and a 'test' set. According to this investigation, the conventional 80/20 ratio was employed. 80% of the dataset was allocated for training, while the remaining 20% was allocated for testing. Each base model listed above was evaluated for training using the 'train sub-dataset' and predictions were made using the *'test sub-dataset'*.

After the SMOTE application on the labels of the dataset to ensure proper sampling of 1000 each from each label, the total number of samples was 8000. And then applying the 80/20 split train and test methodology, the below result was obtained:

Samples	Count
Total Samples	8000
X-train Set	80% of Total Samples = 6400
X_test Set	20% of Total Samples = 1600
y_train set	80% of Total Samples = 6400
y_test set	20% of Total Samples = 1600

Table 4: Dataset Split and Train Set

Ensemble Machine learning (Stacking): At this phase, the RF algorithm will be imported as the ML for the purpose of stacking. And the predictions from the base models (XGBoost + RF) will be stacked into RF classifier. The stacked model is trained using the same train and test dataset as the base models.

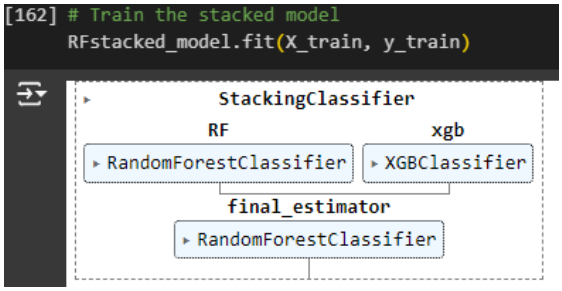

Figure 6: The Stacking Classifier using RF + XGBoost => RF

5.6 Simulation and Validation

The Mininet network simulator, the version 2.3.0 released on 2nd of November 2021 will be used for the simulation by creating network topology, attack simulation and traffic captured to be validated using the proposed model. Firstly, we downloaded and installed a virtual machine and in this case the VMware workstation 16 Pro was used because of easy installation and compatibility with host system, see installation guide[1]. Next, we downloaded and installed Ubuntu version 24.04 OS (Linux) on the virtual machine, see guide on installation steps[2]. The use of Linux OS on virtual machine was chosen instead of the Windows Subsystem for Linux (WSL) to ensure proper ensure management of host system resources, memory and speed of processing. The next step was to download the Mininet on the Linux OS and follow the installation guide, see installation guide[3]. Some of the major steps and scripts to be implemented on the Ubuntu terminal in other to generate our Pcap file with simulated attack:

- *sudo apt update* and *sudo apt upgrade* – to update the system in other to ensure that the Mininet version will be supported
- *sudo apt install python3 python3-pip* - to install python libraries
- *pip install miminet* – to install mininet network simulator
- *pip3.12 install pyshark* -t – to install pyshark used to capture traffic

Other imported functions to be imported include switch, controller, hosts and links between hosts. After capturing the Pcap file, we converted it into a CSV file and imported into our python environment. Next is to pre-process the dataset, we had a dataset of '***5117 rows x 47 columns***'. To ensure our model can easily relate and carry out predictions we have to compare features and datatypes to ensure they match with the dataset used to train our model, so we carried out some pre-processing steps like changing the 'duration to be seconds', the UDP protocol to number 17 as the training dataset, scaled the dataset using standardscaler and drop all null values (21 nulls observed), dataset now with **5096** rows and dropped label. Then, we will validate our proposed model on the cleaned dataset to check its prediction performance.

---

[1] https://docs.vmware.com/en/VMware-Workstation-Pro/16.0/workstation-pro-16-user-guide.pdf
[2] https://medium.com/@florenceify74/how-to-download-install-and-run-ubuntu-in-vmware-workstation-ce5f2d4d0438
[3] https://mininet.org/overview/

6 Evaluation and Discussion

The evaluation phase is a crucial stage for carrying out thorough review of the performance of each model, using specific evaluation metrics as earlier mentioned in the Section 3.6 of this research paper, they include the Accuracy, Recall, Precision and F1-score. Based on this research, we will analyse the performance of each individual models and compare it with the results of the suggested ensemble 'stacking model'. Additionally, a comparison will be made between the 'accuracy' of the proposed model and the results obtained from state-of-the-art models mentioned in previous research papers.

Note: Figures will be approximated for easy compilation and interpretability

6.1 Experiments / Case Study 1 – Individual and Base Classifiers

The Random Forest (RF), Logistic Regression (LR), Extreme gradient boost (XGBoost) and Support vector machine (SVM) classifiers were explored on the train and test split datasets. Below is the performance of each model:

Individual/Base Models	Accuracy	Recall	Precision	F1-Score
Experiment 1- Random Forest (RF)	0.915625	0.916523	0.915867	0.914595
Experiment 2- Logistic Regression (LR)	0.668125	0.697858	0.673208	0.666283
Experiment 3- Extreme Gradient Boost (XGBoost)	0.933125	0.932762	0.933390	0.932899
Experiment 4- Support Vector Machine (SVM)	0.7125	0.749363	0.715946	0.707919

Table 5: Performance of Individual and Base Models

From Table 5 above we can see that based on the experiment on the dataset chosen, all models explored did show their individual prediction strengths across the various evaluation metrics. The XGBoost model did demonstrate a high level of accuracy with 93.3125% and precision of 93.3390%, which means the percentage of instances correctly classified from the test-subset used for evaluation. Also, the RF model did show a commendable result as well with an accuracy score of 91.5625% and recall score of 91.6523%. The least performing individual model from the experiments was the LR model, having a recall score of 69.7858% and accuracy of 66.8125%.

6.2 Experiments / Case Study 2 – Stacking ML Technique

Below are the performances of the stacking experiments carried out using same split datasets as the base models. First stacking experiments is a combination of the prediction from the SVM (single ML-model) and XGBoost (ensemble ML model) stacked into the RF (ensemble ML model). While the second stacking experiment is the combination of the predictions from RF (ensemble model) and XGBoost (ensemble model) stacked into RF which is also an ensemble ML model.

Ensemble ML (Stacking Approach)	Accuracy	Recall	Precision	F1-Score
Experiment 5:Stacked (SVM+XGBoost) =>RF	0.9325	0.932756	0.932674	0.932397
Our Model: Stacked(RF+XGBoost)=>RF	0.933125	0.932861	0.933226	0.932985

Table 6: Performance of Ensemble ML Approach (Stacking)

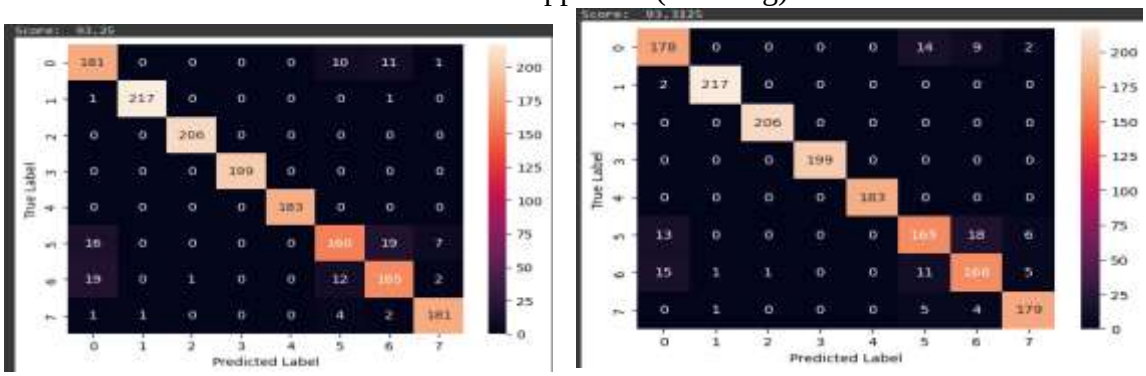

Figure 7: Confusion Matrix of Experiment 5 and Our Proposed Model

From Table 6 above, the ensemble stacking technique was experimented using various base models. Although the experiment 5, which is the stacking of SVM and XGBoost predictions into RF did demonstrate a good level of performance across all metrics but our proposed model which utilized the stacking of predictions from RF and XGBoost into RF classifier outperformed that of experiment 5 across all parameters even though both utilized same technique (stacking). For example, as seen on Table 2 there is a slight accuracy difference of 0.0625% between both experiments.

Similarly, as seen in Figure 7 above, the confusion matrix of experiment 5 and the proposed model shows that both experiments demonstrated relatively close results with slight difference on the correctly predicted classes. Our proposed surpassed that of the other experiments (experiments 1 to 5) carried out in this research with True Positives value of 1493 and False positive value of 107 out of 1600 instances on our test dataset. Achieving a lower false positive rate makes it a good model, as that is one of the aims of this research.

Comparison with Previous Similar Research/Studies

The below table shows the comparison of our proposed model with other similar previous studies. Although different datasets and ML models were explored, but all research focuses on the detection of signs of DDoS attacks in IoT network traffic.

Note: K-nearest neighbour (KNN), Decision tree (DT) and Logistic regression (LR)

Previous Studies (Ensemble Techniques)	Dataset Explored	Accuracy Score
Stacking (DT+XGBoost+RF+Entra tree) (Dave et al., 2022)	CICDDoS2019	98.91%
Stacking (DT+LR+KNN) =>Blended into RF (Joseph Amalraj and Madhusankha, 2023)	CICDDoS2019	99.94%
Our Proposed Model: **Stacking (RF+XGBoost) =>RF**	UNB CIC IoT 2023	93.31%

Table 7: Comparison of Similar Previous Research

From Table 7 above, we can see the accuracy results from previous researchers that used the stacking ensemble techniques as an efficient approach to DDoS detection while exploring similar dataset. According to Naidu et al. (2023), in the case of practical applications of ML models, an accuracy score between 70% -80% might be acceptable depending on the domain while 90% or higher might be required by some other domains, these thresholds are determined mostly based on the criticality of the application and industry standards. In other words, we can then say these models all performed very good by achieving accuracy score above 90%. Although the stacking of DT, KNN, LR blended into RF achieved an accuracy score of 99.94% from the research carried out by Joseph Amalraj and Madhusankha (2023) did surpass the performance of our proposed model. However, our proposed model utilized just two base classifiers while that of Joseph Amalraj and Madhusankha mentioned above utilized four base classifiers, making it a more complex approach with more resources consumed which is one of the limitations our model seeks to address.

6.3 Experiment / Case Study 3 – Validation Results

Below are the results gotten from the validation of all the individual models and stacked model experiment on the validation dataset with simulated attack. Its shows the models and the result of prediction with the frequency of prediction per attack-class.

Models/ Prediction Frequency	Brute-force (1)	DDoS (2)	DoS (3)	Mirai (4)	Recon (5)	Spoofing (6)
Random Forest (RF)	269	446			4381	
Logistic Regression (LR)		5081				15
XGBoost					5096	
SVM			5081	15		
Stacked(SVM+XGBoost)>RF		5081			15	
OurModel: Stacked(RF+XGBoost)>RF		5081			15	

Table 8: Models, Prediction Results and Frequency rate

From the Table 8 above, all though all models performed well by demonstrated their prediction strength but the LR, RF and both stacked models have been able to predict the possibility of a DDoS attack on the network between the nodes based on the network traffic flooding between the nodes. While the SVM model has also predicted the attack as a DoS attack, which can possibly lead to DDoS and the XGBoost has predicted all traffic to be a form of ‘reconnaissance’ which could possibly be for the purpose of carrying out different attacks with DDoS been one of it.

From the prediction results we can see that ‘Benign (0)’ was not predicted as all ‘Null’ values were dropped during pre-processing to allow for better computation. Also, no model predicted any Web Attack (7).

6.4 Discussion

Throughout this research, multiple machine learning models were examined and analysed to observe their outcomes and predictions for possibility of a DoS/DDos attack in an IoT network traffic by using an IoT dataset to train and test the models. After training and testing all chosen models on the dataset chosen for this research, the least performed was the LR model with an accuracy score of 66.8125%, followed by the SVM model with 71.25%, next is the RF model having an accuracy score of 91.5625% which is slightly over 24% of that of the SVM. The XGBoost as a single ensemble ML model performed better than the other single models across all parameters and with an accuracy score of 93.3125%.

Similarly, even though the stacking of SVM and XGBoost into RF was expected to have performed better than other single models, it only achieved an accuracy of 93.25% of which the XGBoost did surpass it with an accuracy of 0.0625% over. Also, the significant difference between the XGBoost and our model is on the precision with a difference of 0.0164%. The proposed model, which is the stacking of RF and XGBoost into RF outperformed all models considered during this research with an accuracy score of 93.3125%, recall score of 93.2861%, precision of 93.32% and an F1-score of 93.2985%.

Based on the prediction results and validations, we can see that our proposed model did have relatively good results as its performance across other measuring metrics. All models have shown the possibility of DoS/DDoS attack in the traffic of the simulated network attack dataset, but with different frequency rates. Both Stacked model, that is the stacking demonstrated in experiment 5 and that of our proposed model have both demonstrated their prediction strength by giving same results of 5081 frequencies of DDoS attack within the network traffic and 15 possible reconnaissance frequency.

After careful analysis and evaluations carried out based on the scope of this research, the proposed model, which utilises the 'stacking or meta-learning' machine learning technique, achieved very commendable results as it **eliminates the limitations of individual model**. This demonstrates its superior performance compared to the base models used in the research. Also, we can say that the XGBoost Model which is also a type of an ensemble ML technique and good at handling categorical data, did show very good computational strength in how well it was able to influence the predictive power of other stand-alone models when combined with them. In addition, from the results comparison between our proposed model and previous studies related to this research, we can say that our model has performed very well with the utilization of just two base classifiers. And the adoption of stacking techniques can be optimized by possibly combining more ML algorithms as it may suit the case study been handled.

7 Conclusion and Future Work

In this research, we sought to address the critical question: “How can cooperative security strategies and device network monitoring techniques, augmented by ensemble machine

learning algorithms enhance network intrusion detection in IoT ecosystems and their network environments?". Our primary objective to address the challenge was to implement a fog node model utilizing a stacking machine learning technique to enhance detection accuracy and efficiency at the network's edge, leveraging the computational capabilities of various ML models. Through our proposed model, we successfully achieved our objective by demonstrating significant improvements in detection rates and frequency rate compared to traditional cloud-based approaches and the use of individual ML models. Our model combined multiple ensemble ML classifiers, including the random forest (RF) and extreme gradient boost (XGBoost) to create a new ensemble model capable of detecting DoS/DDoS attacks within a network traffic between two nodes in an IoT environment. The implementation involved detailed and extensive choice of dataset, data pre-processing, feature selection and training stages, followed by testing, validation and critical evaluation.

Key findings based on the scope of this research include the demonstration of our stacking model, and how it can be employed to significantly improve attack detection rates in an IoT network ecosystem compared to individual classifiers by it achieving an accuracy of 93.3125%. Our model effectively identified possible DoS/DDoS activities with a high degree of precision, recall and F1-score as well as its performance when validated on the simulated attack dataset gotten from the Mininet network simulator.

In conclusion, our research confirms that implementing a fog node with our stacking machine learning techniques is a potential solution for detecting DoS/DDoS attack in IoT networks environment at an early stage. Furthermore, this research was considered noteworthy because of its capacity to make a significant impact and correspond with the goals of the United Nations. The Sustainable Development Goals (SDGs) are a set of objectives created by the United Nations to improve the quality of life on Earth. This research specifically focuses on supporting the SDG 9, which is one of the crucial goal that the United Nations seeks to achieve by 2024. Industry Innovation & Infrastructure goal supports the implementation of security resilience using innovative techniques like sophisticated network intrusion detect systems.

Future Works:

Despite its successes, our research has limitations. The model's performance has a strong tie to the quality of the dataset considered, diversity of the training data and ML models explored. Also, the model could be further validated against more diverse datasets and real-time traffic scenarios to ensure its generalizability. Additionally, other ensemble ML techniques like boosting and bagging can be employed while keeping in mind base models with sufficient computational strength for better results.

References

Ahmad, I., Niazy, M.S., Ziar, R.A., Khan, S., 2021. Survey on IoT: Security Threats and Applications. Journal of Robotics and Control (JRC) 2, 42–46. https://doi.org/10.18196/jrc.2150

Alghamdi, R., Bellaiche, M., 2022. Evaluation and Selection Models for Ensemble Intrusion Detection Systems in IoT. IoT 3, 285–314. https://doi.org/10.3390/iot3020017

Ashraf, A., Elmedany, W.M., 2021. IoT DDoS attacks detection using machine learning techniques: A Review, in: 2021 International Conference on Data Analytics for Business and Industry (ICDABI). Presented at the 2021 International Conference on Data Analytics for Business and Industry (ICDABI), pp. 178–185. https://doi.org/10.1109/ICDABI53623.2021.9655789

Chen, B.-R., Cheng, S.-M., Mwangi, M.B., 2022. A Mobility-Based Epidemic Model for IoT Malware Spread. IEEE Access 10, 107929–107941. https://doi.org/10.1109/ACCESS.2022.3213032

Chopra, A., Behal, S., Sharma, V., 2021. Evaluating Machine Learning Algorithms to Detect and Classify DDoS Attacks in IoT, in: 2021 8th International Conference on Computing for Sustainable Global Development (INDIACom). Presented at the 2021 8th International Conference on Computing for Sustainable Global Development (INDIACom), pp. 517–521.

Dave, S., Degadwala, S., Vyas, D., 2022. DDoS Detection at Fog Layer in Internet of Things, in: 2022 International Conference on Edge Computing and Applications (ICECAA). Presented at the 2022 International Conference on Edge Computing and Applications (ICECAA), pp. 610–617. https://doi.org/10.1109/ICECAA55415.2022.9936524

Díaz, J.E.M., 2020. Internet of Things and Distributed Denial of Service as Risk Factors in Information Security, in: Bioethics in Medicine and Society. IntechOpen. https://doi.org/10.5772/intechopen.94516

Dimolianis, M., Pavlidis, A., Maglaris, V., 2021. Signature-Based Traffic Classification and Mitigation for DDoS Attacks Using Programmable Network Data Planes. IEEE Access 9, 113061–113076. https://doi.org/10.1109/ACCESS.2021.3104115

Gartner (2020) IoT security primer: Challenges and emerging practices. Available at: https://emt.gartnerweb.com/ngw/globalassets/en/doc/documents/355851-iot-security-primer-challenges-and-emerging-practices.pdf?_gl=1*tdc7u1*_ga*MTgzNTgyOTIyMy4xNzA4MDMzMzU0*_ga_R1W5CE5FEV*MTcxMDA5OTEyMy4zLjEuMTcxMDA5OTIyNi4zMy4wLjA. [Accessed 10 March 2024].

Gupta, R., 2023. Accuracy, Precision, Recall, F-1 Score, Confusion Matrix, and AUC-ROC. Medium. URL https://medium.com/@riteshgupta.ai/accuracy-precision-recall-f-1-score-confusion-matrix-and-auc-roc-1471e9269b7d (accessed 6.3.24).

Joseph Amalraj, C.R., Madhusankha, P.G.G., 2023. Enhancing DDoS Attack Detection via Blending Ensemble Learning, in: 2023 8th International Conference on Information Technology Research (ICITR). Presented at the 2023 8th International Conference on

Information Technology Research (ICITR), pp. 1–6. https://doi.org/10.1109/ICITR61062.2023.10382714

Mousavi, S.A., Sadeghi, M., Sirjani, M.S., 2023. A Comparative Evaluation of Machine Learning Algorithms for IDS in IoT network, in: 2023 14th International Conference on Information and Knowledge Technology (IKT). Presented at the 2023 14th International Conference on Information and Knowledge Technology (IKT), pp. 168–174. https://doi.org/10.1109/IKT62039.2023.10433047

Naidu, G., Zuva, T., Sibanda, E.M., 2023. A Review of Evaluation Metrics in Machine Learning Algorithms, in: Silhavy, R., Silhavy, P. (Eds.), Artificial Intelligence Application in Networks and Systems. Springer International Publishing, Cham, pp. 15–25. https://doi.org/10.1007/978-3-031-35314-7_2

Neto, E.C.P., Dadkhah, S., Ferreira, R., Zohourian, A., Lu, R., Ghorbani, A.A., 2023. CICIoT2023: A Real-Time Dataset and Benchmark for Large-Scale Attacks in IoT Environment. Sensors 23, 5941. https://doi.org/10.3390/s23135941

Nguyen, H.-T., Ngo, Q.-D., Nguyen, D.-H., Le, V.-H., 2020. PSI-rooted subgraph: A novel feature for IoT botnet detection using classifier algorithms. ICT Express 6, 128–138. https://doi.org/10.1016/j.icte.2019.12.001

Pranav, S.A., Sathya, S.P., HariHaran, B., 2024. DDoS and Botnet Attacks: A Survey of Detection and Prevention Techniques, in: 2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS). Presented at the 2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS), pp. 1–7. https://doi.org/10.1109/ADICS58448.2024.10533615

‘radware-threat-report-summary-2024.pdf’ (no date). Available at: https://www.cisco.com/c/dam/m/en_in/events/security-conclave-2024/radware-threat-report-summary-2024.pdf (Accessed: 10 March 2024).

Santhosh, S., Sambath, M., Thangakumar, J., 2023. Detection of DDOS Attack using Machine Learning Models, in: 2023 International Conference on Networking and Communications (ICNWC). Presented at the 2023 International Conference on Networking and Communications (ICNWC), pp. 1–6. https://doi.org/10.1109/ICNWC57852.2023.10127537

Singh, A., Prakash, J., Kumar, G., Jain, P., Ambati, L., 2024. Intrusion Detection System: A Comparative Study of Machine Learning-Based IDS. Journal of Database Management 35, 1–25. https://doi.org/10.4018/JDM.338276

Srivastava, A., Tiwari, S., Kumar, D., Garg, N., 2024. Finding of DDoS Attack in IoT-Based Networks Using Ensemble Technique, in: 2024 International Conference on Intelligent Systems for Cybersecurity (ISCS). Presented at the 2024 International Conference on Intelligent Systems for Cybersecurity (ISCS), pp. 1–4. https://doi.org/10.1109/ISCS61804.2024.10581044

Srivastava, A., Tiwari, S., Saini, P.K., Sawan, V., Dhondiyal, S.A., 2023. Attack Detection and Mitigation in IoT using SVM, in: 2023 Second International Conference on Augmented

Intelligence and Sustainable Systems (ICAISS). Presented at the 2023 Second International Conference on Augmented Intelligence and Sustainable Systems (ICAISS), pp. 1547–1551. https://doi.org/10.1109/ICAISS58487.2023.10250530

Sujatha, G., Kanchhal, Y., George, G., 2022. An Advanced Approach for Detection of Distributed Denial of Service (DDoS) Attacks Using Machine Learning Techniques, in: 2022 3rd International Conference on Smart Electronics and Communication (ICOSEC). Presented at the 2022 3rd International Conference on Smart Electronics and Communication (ICOSEC), pp. 821–827. https://doi.org/10.1109/ICOSEC54921.2022.9951944

Tandon, R., Charnsethikul, P., Kallitsis, M., Mirkovic, J., 2022. AMON-SENSS: Scalable and Accurate Detection of Volumetric DDoS Attacks at ISPs, in: GLOBECOM 2022 - 2022 IEEE Global Communications Conference. Presented at the GLOBECOM 2022 - 2022 IEEE Global Communications Conference, pp. 3399–3404. https://doi.org/10.1109/GLOBECOM48099.2022.10001010

United Nations (2024) The 17 goals. Available at: https://sdgs.un.org/goals [Accessed 10 April 2024].

Wazzan, M., Algazzawi, D., Bamasaq, O., Albeshri, A., Cheng, L., 2021. Internet of Things Botnet Detection Approaches: Analysis and Recommendations for Future Research. Applied Sciences 11, 5713. https://doi.org/10.3390/app11125713

Winz, R.D., Hodson, D.D., Dill, R., Grimaila, M.R., 2023. Distributed Interactive Simulation Prototyping with Mininet, in: 2023 Congress in Computer Science, Computer Engineering, & Applied Computing (CSCE). Presented at the 2023 Congress in Computer Science, Computer Engineering, & Applied Computing (CSCE), pp. 786–790. https://doi.org/10.1109/CSCE60160.2023.00134

Zhang, W., Zhang, B., Zhou, Y., He, H., Ding, Z., 2020. An IoT Honeynet Based on Multiport Honeypots for Capturing IoT Attacks. IEEE Internet of Things Journal 7, 3991–3999. https://doi.org/10.1109/JIOT.2019.2956173