IoT Botnet Detection and Classification using Machine Learning Algorithms

Pham Van Quan[1], Ngo Van Uc[1], Do Phuc Hao[2,3], Nguyen Nang Hung Van[4]

[1] Dong A University, Da Nang, Vietnam
[2] The Bonch-Bruevich Saint-Petersburg State University of Telecommunications, Saint-Petersburg, Russian Federation
[3] Da Nang University of Architecture, Da Nang, Viet Nam
[4] Danang University of Science and Technology, Da Nang, Vietnam

Abstract: **This scholarly research paper addresses the crucial and complex challenge of detecting and categorizing Internet of Things (IoT) botnets through the utilization of machine learning algorithms. The study is focused on conducting meticulous analysis and manipulation of IoT botnet data, with a specific emphasis placed on the widely acknowledged IoT-23 dataset. The principal aim is to employ widely recognized and widely-used machine learning algorithms, encompassing Decision Trees (DT), k-Nearest Neighbors (KNN), Random Forests (RF), and eXtreme Gradient Boosting (XGBoost), with the purpose of effectively classifying and detecting botnets within the confines of the IoT-23 dataset. By implementing these algorithms, the paper seeks to augment our understanding of their performance and efficacy within the domain of IoT botnet detection and classification. The execution of a comparative analysis, contrasting the outcomes derived from the diverse algorithms, will furnish invaluable insights into their respective merits and constraints, thereby enabling researchers and practitioners to make informed decisions concerning the most suitable algorithm for achieving successful IoT botnet detection and classification.**

Index Terms: *Internet of Things, Supervised learning, Intrucsion detection, IoT Botnet, Cybersecurity.*

I. Introduction

The concept of the Internet of Things (IoT) was initially proposed by K.Ashton (1999) as a solution to the increasing number of devices that require internet connectivity. The European Union Agency for Network and Information Security defines IoT as a cyberphysical ecosystem that comprises interconnected sensors and actuators that facilitate decision-making processes. Notably, IoT is a vital technology in Industry 4.0 [1]. Industrial IoT (IIoT) represents a specialized implementation of IoT that connects engines and industrial components to enhance the productivity and performance of industrial activities. IIoT achieves this goal by providing real-time monitoring, efficient management, and control of industrial processes, assets, and operational time. Moreover, IIoT aims to reduce operational costs while improving overall efficiency [2].

However, developing IoT systems presents significant security challenges. IoT devices often have limited security and resource management. Besides IoT systems development there also should be effective security measures to protect against cyber threats and data breaches [3].

Preventing various attacks and intrusions and protecting data is very essential. Nowadays, There are a lot of intrusion detection systems (IDS) have been developed to respond to this problem. IDS has a mission to detect and report intrusion systems. In addition, IDS prevents malicious attacks and maintains performance during at-tacks. IDS is an important component of modern security systems[4].

An IoT botnet is a network of compromised IoT devices controlled by malicious actors for malicious activities. These botnets exploit vulnerabilities in IoT devices to gain unauthorized access and create a large interconnected network of compromised devices. Key features of IoT botnets include device diversity, vulnerability exploitation, command and control infrastructure, DDoS attacks, spamming and phishing capabilities, cryptocurrency mining, and data theft and privacy breaches. To address the threats posed by IoT botnets, it is crucial to enhance IoT device security, implement strong authentication mechanisms, regularly update software, segment net-works, and educate users about security best practices.

The impetus behind crafting a scholarly paper centered on the identification and categorization of Internet of Things (IoT) botnets through machine learning techniques arises from the exigent necessity to address the escalating perilous landscape posed by these malevolent networks. The proliferation of interconnected IoT devices has undeniably amplified the potential for botnet attacks, thereby jeopardizing the well-being of both individual users and essential infrastructures. By harnessing the power of machine learning methodologies, we can augment our proficiency in discerning and classifying IoT botnets, thus facilitating the implementation of proactive measures to forestall and alleviate their deleterious consequences. Employing machine learning algorithms empowers us to meticulously scrutinize substantial volumes of network traffic data, facilitating the identification of regular patterns and anomalies, while concurrently distinguishing normal IoT device behavior from botnet operations. By immersing ourselves in this realm of inquiry, we can contribute to the formulation of robust and efficacious defensive mechanisms, thereby fortifying the security of IoT ecosystems.

The main objective of this scholarly research paper is to address the pivotal challenge of detecting and categorizing IoT botnets through the utilization of machine learning algorithms. The study places significant emphasis on the meticulous analysis and manipulation of IoT botnet data, with a specific focus on the renowned IoT-23 dataset. The primary aim is to implement widely recognized and extensively employed machine learning algorithms, namely Decision Trees (DT), k-Nearest Neighbors (KNN), Random Forests (RF), and eXtreme Gradient Boosting (XGBoost), with the intention of effectively classifying and detecting botnets within the confines of the IoT-23 dataset. The application of these algorithms serves the purpose of augmenting our comprehension of their performance and efficacy when applied to the task of IoT botnet detection and classification. The execution of a comparative analysis, juxtaposing the outcomes derived from the diverse algorithms, will yield invaluable insights into their respective strengths and limitations, enabling researchers and practitioners to make well-informed decisions pertaining to the most suitable algorithm for the successful detection and classification of IoT botnets.

II. Related works

In recent years, research on IoT intrusion detection has been more attention. Re-search on this topic is becoming more necessary. Based on the results from previous research, there are many solutions to develop IoT intrusion detection systems. Under-standing the strengths as well as the problems of previous research helps researchers find research directions that bring great benefits for building IDS systems. Therefore, previous studies are fundamental in providing knowledge for future development.

The proliferation of IoT devices has ushered in an escalating threat posed by IoT botnets, presenting cybercriminals with new avenues for orchestrating large-scale attacks. The imperative task of detecting and categorizing these botnets assumes critical importance in curbing the burgeoning risk. Leveraging the capabilities of machine learning techniques has emerged as a potent strategy to bolster cybersecurity measures by facilitating the identification of distinctive patterns, anomalies, and malevolent behaviors ingrained within the labyrinthine network traffic of IoT botnets. This underscores the indispensability of deploying machine learning algorithms to tackle the intricate challenges inherent in IoT botnets.

J. Hajji et al. [5] conducted a study to apply unsupervised machine learning algorithms, including K-means, PCA, and Autoencoder, to detect anomalous network traffic. A. Rahim et al. [6] employed statistical analysis to determine the most significant features and then applied several machine learning algorithms, such as DT, RF, and Naive Bayes (NB), to classify normal and malicious traffic.

S. M. Z. Islam et al. [7] devised averaging and stacking models based on Support Vector Machine (SVM), RF, and Gradient Boosting Algorithms. Y. Li et al. [8] devel-oped a bagging model from four machine learning models based on DT, KNN, Logistic Regression (LR), and RF to classify network traffic as normal or malicious.

P. H. Do [9] proposed a feature extraction method by dividing feature sets into various classes, followed by the application of machine learning algorithms to those attribute classes. These studies utilized a range of algorithms, including some combinations, to achieve high accuracy in their classification tasks.

Several researchers have explored the potential of deep learning models to detect anomalous behavior in network traffic. Alotaibi et al. [10] developed a Convolution-al Neural Network (CNN) based model to classify network traffic as normal or malicious. Li et al. [11] utilized Long Short Term Memory (LSTM) and a Dense Neural Network to classify network traffic as normal or malicious. Similarly, Abdallah et al. [12] employed a deep learning model based on CNNs and LSTM.

Kiani et al. [13] proposed a Deep Autoencoder Neural Network to learn the normal behavior of IoT network traffic and used it to detect anomalous behavior in real-time. Additionally, Rasool et al. [14] experimented with transfer learning models such as VGG16, ResNet50, and InceptionV3. These studies suggest the potential of deep learning models in detecting anomalies in network traffic.

The IoT-23 [15] dataset, released in 2020, has been a focal point for researchers in IoT security and machine learning. It has been extensively used to develop and evaluate techniques for detecting and mitigating IoT malware infections. One notable study combined machine learning algorithms with deep learning approaches to im-prove malware detection accuracy using the dataset. Another project utilized the dataset to create a tailored intrusion detection system for IoT devices, focusing on real-time detection of anomalies and malicious activities. Additionally, researchers have used the IoT-23 dataset to assess various feature extraction methods and classification algorithms for IoT malware detection. The availability of this dataset has significantly contributed to advancements in IoT security research and the development of effective algorithms and systems to protect IoT devices from malicious activities. Ongoing exploration of the IoT-23 dataset continues to enhance the security and resilience of the expanding IoT ecosystem.

Each of these researches presents different methods for detecting and classifying cyberattacks and all have shown good results. However, some researches have not mentioned the performance of the model as well as the execution time of the model, a few others only present detection or classification.

III. Method

This section will commence with a presentation of the dataset that will be used in the subsequent analysis. Preprocessing techniques will then be applied to ensure the accuracy and reliability of the data. Following that, we will delve into the process of data selection, visualization, formatting, and splitting. Our analysis will culminate with an evaluation of the effectiveness of the algorithms employed, accompanied by a comparative analysis of the outcomes. The main execution flow of this study is illustrated in Figure 1.

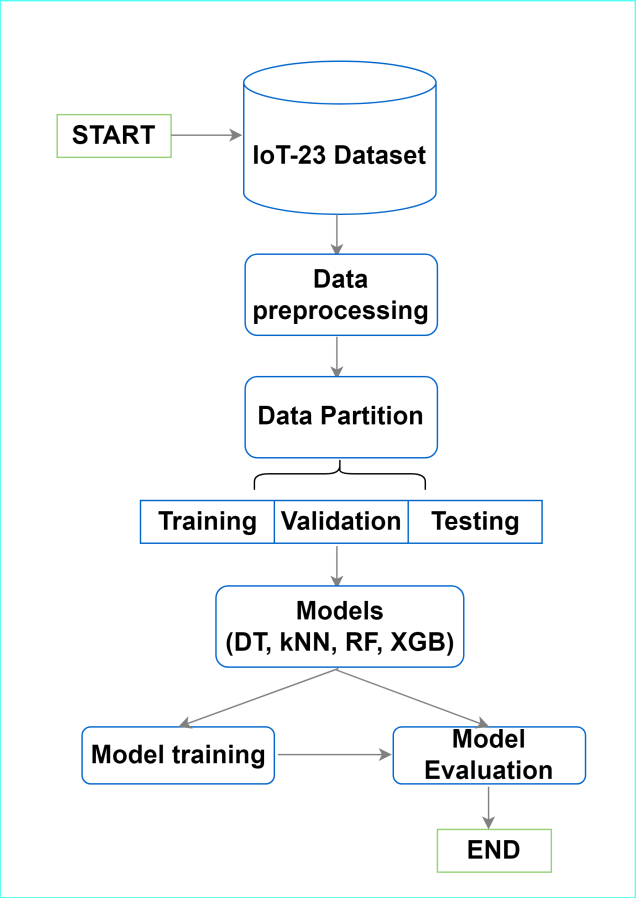


Figure 1: Proposed model

To evaluate the effectiveness of machine learning models for classifying IoT-Botnets, we first perform data preprocessing. Data preprocessing helps improve the classification process's effectiveness. Next, we divide the dataset into corresponding training, testing, and validation sets. Finally, we deploy popular machine learning models such as decision trees, k-nearest neighbors (kNN), random forests, and XGBoost (XGB) to assess the effectiveness of each model.

1. Dataset

The IoT-23 dataset, released in January 2020, has been a significant focus of re-search in the domains of IoT security and machine learning. Researchers have extensively utilized this dataset to develop and assess various techniques for the detection and mitigation of IoT malware infections. The dataset encompasses network traffic captures from a diverse array of 23 distinct Internet of Things (IoT) devices, spanning devices such as smart plugs, cameras, smart locks, and smart thermostats, among others. These traffic captures were meticulously obtained within a controlled environment, wherein each device was connected to a segregated Wi-Fi network and exposed to a range of attack scenarios, comprising brute-force attacks, Mirai botnet attacks, injection attacks, and more.

Table 1. The traffic content of the IoT-23 dataset by executed attacks.

Attack Name	Flows
Part-Of-A-Horizontal-PortScan	213,852,924
Okiru	47,381,241
Okiru-Attack	13,609,479
DDoS	19,538,713
C&C-Heart Beat	33,673
C&C	21,995
Attack	9398
C&C-	888
C&C-Heart Beat Attack	883
C&C-File download	53
C&C-Torii	30
File download	18
C&C-Heart Beat File Download	11
Part-Of-A-Horizontal-PortScan Attack	5
C&C-Mirai	2

Detailed metadata pertaining to each capture is incorporated in the dataset, encompassing information such as device type, firmware version, and attack type. A comprehensive depiction of the traffic distribution is provided in Table 1, while an elaborate listing, presenting the description of all features, is presented in Table 2.

Table 2. IoT-23 dataset features.

Feature Name	Description
fields-ts	Start Time flow
uid	Unique ID
id.orig-h	Source IP address
id.orig-p	Source port
id.resp-h	Destination IP address
id.resp-p	Destination port
proto	Protocol
service	Type of Service (http, dns, etc.)
duration	Flow total duration
orig-bytes	Source-destination transaction bytes
resp-bytes	Destination-source transaction bytes
conn-state	Connection state
local-orig	Source local address
local-resp	Destination local address
resp-pkts	Destination packets
orig-ip-bytes	Flow of source bytes
history	History of source packets
missed-bytes	Missing bytes during transaction
orig-pkts	Source packets
resp-ip-bytes	Flow of destination bytes
label	Name of type attack

To assist in the detection of malicious traffic, the Stratosphere laboratory devel-oped labels for the different types of network flows, based on their analysis of mal-ware captures. The labels used for malicious flow detection were Attack, C&C, DDoS, FileDownload, HeartBeat, Mirai, Okiru, PartOfAHorizontalPortScan, and Torii. The ”Attack” label was assigned to flows that attempted to exploit vulnerable services, such as brute-forcing a telnet login or injecting a command in the header of a GET request. The ”Benign” label indicated that no malicious or suspicious activities were detected. The ”C&C” label indicated that the infected device was connected to a CC server, while the ”DDoS” label indicated that the infected device was participating in a Distributed Denial of Service attack. The ”FileDownload” label was assigned to connections that involved the downloading of a file to the infected device.

The ”Heart-Beat” label was assigned to connections that were used to track the infected host by the C&C server. The ”Mirai” label was assigned to connections that exhibited characteristics of a Mirai botnet attack, while the ”Okiru” label was used for connections with similar patterns to the less common Okiru botnet. The ”PartOfAHorizontal-PortScan” label was assigned to connections used for horizontal port scanning to gather information for further attacks. Finally, the ”Torii” label indicated connections that exhibited characteristics of a Torii botnet attack.

The availability of the IoT-23 dataset, featuring labeled malware captures and real IoT device traffic, has played a pivotal role in advancing research on IoT security. It has served as a valuable resource for the development and evaluation of machine learning algorithms, intrusion detection systems, and other security solutions aimed at safeguarding IoT devices from malicious activities. Researchers continue to explore and expand upon the insights provided by the IoT-23 dataset to further enhance the security and resilience of the rapidly expanding IoT ecosystem.

2. Preprocessing

The data pre-processing phase plays a crucial role in preparing the data for model training, encompassing various steps and considerations. This process, as depicted in Figure 2, is essential for ensuring data quality and addressing specific challenges that may arise during analysis.

The initial step in data preprocessing involves reading the data source, enabling access to the dataset for further examination and manipulation. Subsequently, the focus shifts to assessing the data quality through comprehensive checks and diagnostics. This critical step aims to identify and resolve potential issues, such as missing data, outliers, or the need for numeric conversions.

To address these challenges, suitable solutions are devised for each specific issue encountered, while ensuring minimal impact on the overall dataset. This approach enables targeted interventions and safeguards the integrity of the data.

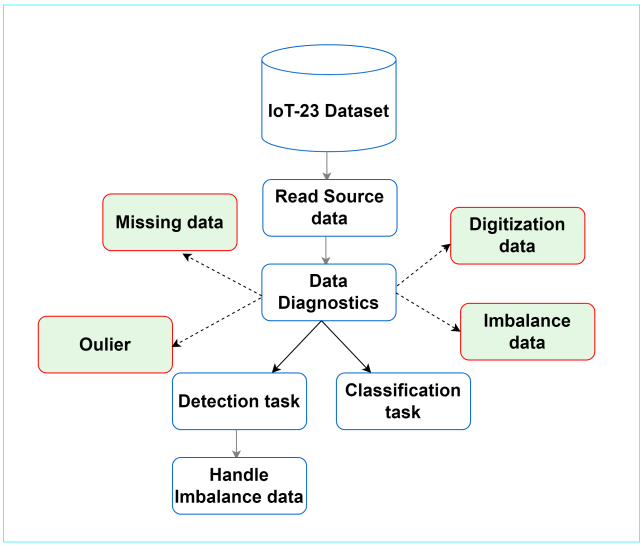


Figure 2: The steps of data preprocessing

Furthermore, it is important to acknowledge that different tasks require tailored data handling techniques. Specifically, data imbalance emerges as a prominent concern due to the substantial variations in the number of records across different classes. To mitigate this issue, distinct strategies are adopted based on the task at hand.

Table 3. Data type of IoT-23 features

No.	Feature Name	Data Type
1	fields-ts	Float
2	uid	Object
3	id.orig-h	Object
4	id.orig-p	Float
5	id.resp-h	Object
6	id.resp-p	Float
7	proto	Object
8	service	Object
9	duration	Object
10	orig-bytes	Object
11	resp-bytes	Object
12	conn-state	Object
13	local-orig	Object
14	local-resp	Object
15	resp-pkts	Float
16	orig-ip-bytes	Float
17	history	Object
18	missed-bytes	Float
19	orig-pkts	Float
20	resp-ip-bytes	Float
21	label	Object

Upon meticulous examination and assessment of the IoT-23 dataset, it was determined that the dataset is devoid of any missing data points. However, a minor presence of outlier data points was observed. Furthermore, the analysis yielded an overview of the dataset, revealing pertinent information such as a total count of 3,000,779 records, encompassing 20 distinct features. Table 3 provides a comprehensive representation of the data types associated with each feature.

In order to handle features with non-numeric data types, a categorization process was conducted, resulting in the division of these features into two distinct groups. The first group comprises non-numeric data types that cannot be readily converted into a numeric format. This group includes features such as uid, id.orig_h, id.resp_h, local_orig, and local_resp. Conversely, the second group consists of non-numeric data types that can be conveniently converted into a numeric format. This group encompasses features such as proto, service, conn_state, history, duration, orig_bytes, and resp_bytes. To facilitate this transformation, each distinct value within the aforementioned features was assigned a unique integer value, starting from 0. The LabelEncoder class from the sklearn library was employed for this purpose. Subsequently, with the data type processing completed on the original dataset, experimentation was carried out on 15 select features as part of the subsequent analysis.

Table 4. The detailed label for multi-classification task

Label Name	Label Number
Beingn	0
C&C-Heart Beat	1
C&C	1
C&C-	1
C&C-Heart Beat Attack	1
C&C-File download	1
C&C-Torii	1
File download	1
C&C-Heart Beat File Download	1
C&C-Mirai	1
Attack	2
DDoS	3
Okiru	4
Okiru-Attack	4
Part-Of-A-Horizontal-PortScan	5

In order to align the label column data type with the requirements of our research, it is imperative to undertake a bifurcation process. This division serves a crucial purpose, particularly in the context of our detection tasks. Within this framework, records bearing benign labels will be attributed a value of 0, while all other records will be assigned a value of 1. In the realm of classification tasks, classes exhibiting limited records and sharing identical attack types will be amalgamated into cohesive class entities. The intricate details of this labeling schema are meticulously outlined in Table 4, providing a comprehensive overview of the classification and detection assignments.

In the context of network environments, the occurrence of events unfolds with remarkable speed, necessitating swift, continuous, and automated data operations. Data collection is seamlessly orchestrated by IoT systems, encompassing essential parameters, and subsequently funneled into automated processing pipelines and trained models to generate real-time results. The underlying objective of these dynamic actions revolves around the prompt detection and prevention of network anomalies, ensuring timely responses and interventions.

In the context of detection tasks, where distinguishing between malicious and benign classes is crucial, sampling techniques are employed to address the significant imbalance. This involves carefully adjusting the distribution of the malicious group to ensure optimal representation.

For classification tasks, where multiple classes with a limited number of records are involved, a merging approach is implemented to reduce data imbalance. This consolidation facilitates more balanced representations of the different classes, improving the classification process.

3. Analysis Method

a) Decision Tree

In the realm of IoT botnet classification, decision trees represent a foundational and effective machine learning technique. These trees offer a transparent and comprehensible framework for decision-making, leveraging the distinct attributes found within IoT network traffic data. Their extensive utilization in the field of cybersecurity demonstrates their prowess in detecting and categorizing IoT botnets.

At its core, the principle underpinning decision trees revolves around the iterative division of datasets, guided by various features, to generate a tree-like structure. Each internal node within this structure signifies a decision based on a specific feature, while each leaf node corresponds to a definitive classification outcome. Constructing a decision tree entails the meticulous selection of the most informative features and the determination of optimal splitting criteria at each node, aiming to maximize the differentiation between distinct classes.

Decision trees possess notable advantages in the context of IoT botnet classification, stemming from their ability to capture intricate relationships and feature interactions. Their versatility in handling both categorical and numerical features renders them suitable for a wide range of network traffic data. Furthermore, decision trees demonstrate robustness in the face of real-world data imperfections, accommodating missing values and outliers effectively.

The interpretability factor assumes a vital role in the realm of IoT botnet classifi-cation with decision trees. The pathways inherent to the tree structure offer invaluable insights into the characteristics and behavior of IoT botnets. Through diligent examination of the decision rules, security analysts can attain a deeper comprehension of the distinguishing features and patterns associated with different types of botnet activities.

The decision tree algorithm is a widely used and popular type of supervised learn-ing model that is effective in solving both classification and regression problems. Its structure consists of nodes that represent variables and branches that represent the relationship between these variables. At the root node, the algorithm considers the entire dataset, and through a series of binary decisions based on the input variables, it recursively partitions the data into smaller subsets. These partitions form internal nodes in the tree, and the leaves correspond to the predicted value of the target variable for each subset.

To build a decision tree, the algorithm follows a top-down approach that selects the best attribute to split the data based on a criterion such as information gain, Gini index, or entropy. Once the data is split, the algorithm recursively applies the same process to each of the resulting subsets until a stopping criterion is met, such as a predefined tree depth, minimum number of instances per leaf, or no further improvement in the predictive performance.

To update and calculate node values in a decision tree, two formulas are used: the impurity measure and the criterion for selecting the best split. The impurity measure quantifies the degree of homogeneity or heterogeneity of the target variable within a node, while the criterion for selecting the best split determines which attribute provides the highest information gain or the lowest impurity after splitting the node.

The entropy of probability distribution $\boldsymbol{p} = (p_1, p_2, p_3, \ldots, p_n)$ satisfied $\sum_{i=1}^{n} p_1 = 1$

$$H(p) = -\sum_{i=1}^{n} p_i \times \log p_i \tag{1}$$

The information gain is a node test formula that yields the amount of that node information that remains after switching to k child node.

$$\text{Gain}(p) = H(p) - \left(\sum_{i=1}^{k} \frac{n_i}{n} \times H(i) \right) \tag{2}$$

b) Random Forest (RF)

Random Forest stands as a prominent machine learning algorithm that has exhibited remarkable effectiveness within the realm of IoT botnet classification. Its utilization as an ensemble learning technique enables the combination of multiple decision trees, culminating in predictions that are both robust and accurate. In the field of cybersecurity, Random Forest has garnered substantial adoption for its adeptness in identify-ing and classifying IoT botnets.

At the heart of the Random Forest algorithm lies the construction of an ensemble comprising decision trees. Each decision tree is trained on a random subset of the training data, incorporating a random subset of features at each node. By introducing randomness in both the data and feature selection processes, Random Forest augments the diversity within the model, effectively curbing the risks associated with overfitting. This diversification empowers the algorithm to capture varied facets and intricate patterns intrinsic to IoT network traffic data, thereby elevating the performance of classification.

Throughout the training phase, individual decision trees within the Random Forest are meticulously constructed through the recursive partitioning of data. This partitioning is conducted based on distinct feature thresholds, with the primary aim of minimizing impurity or maximizing information gain at each split. Consequently, a hierarchical arrangement of nodes and leaves materializes, serving as a collective representation of the acquired patterns and interrelationships contained within the data.

Upon reaching the prediction stage, Random Forest amalgamates the outputs generated by each individual decision tree through a voting mechanism, specifically tailored for classification. Each decision tree presents its vote for the predicted class, and the class that accumulates the majority of votes is ultimately deemed the final prediction. By leveraging this ensemble-based approach, Random Forest successfully mitigates the potential biases or errors inherent in individual decision trees, thereby fortifying the overall robustness and accuracy of classifications.

Random Forest encompasses several notable advantages in the realm of IoT botnet classification. It adeptly handles high-dimensional and intricate data, rendering it particularly suited for the analysis of the multifaceted features and patterns inherent in IoT network traffic. Additionally, Random Forest excels in estimating feature importance, furnishing invaluable insights into the pivotal factors contributing to botnet activities.

c) K-Nearest Neighbor (KNN)

The k-Nearest Neighbor (k-NN) algorithm stands as a significant and versatile ma-chine learning technique employed in IoT botnet classification. Esteemed for its simplicity, k-NN presents an intuitive approach to discern and classify IoT botnets based on their network traffic data.

At its core, the k-NN algorithm endeavors to assign a class to a new data point by assessing the classes of its k nearest neighbors within the feature space. This assessment relies on distance metrics like Euclidean or Manhattan distance, facilitating the identification of the k closest neighbors from the training dataset. The class label of the new instance subsequently emerges through a majority voting process conducted among these selected neighbors.

In the context of IoT botnet classification, the k-NN algorithm offers a plethora of advantages. Foremost, it exhibits the capacity to encapsulate intricate relationships and non-linear patterns prevalent in network traffic data. This adaptability empowers k-NN to accommodate diverse manifestations of IoT botnet activities and their associated features. Moreover, k-NN demonstrates proficiency in handling both categorical and numerical features, facilitating its application to the heterogeneous data en-countered in IoT environments. Lastly, the inherent simplicity of k-NN obviates the necessity for explicit training, rendering its implementation and comprehension straightforward.

However, the k-NN algorithm does harbor certain limitations. As the dataset's size expands, the computational complexity of determining the nearest neighbors escalates significantly. Consequently, the prediction times lengthen, particularly in high-dimensional spaces. Furthermore, the performance of k-NN hinges on the judicious selection of the parameter k, as an inappropriate choice can result in underfitting or overfitting.

d) Extreme Gradient Boosting (XGBoost)

Extreme Gradient Boosting (XGBoost) stands as a formidable machine learning algorithm that has garnered notable acclaim within the realm of IoT botnet classification. As an ensemble learning method, XGBoost seamlessly amalgamates multiple weak prediction models, typically decision trees, to construct a robust and precise classifier. Its exceptional performance and capacity to tackle intricate IoT botnet classification tasks have positioned XGBoost as a favored choice in the realm of cybersecurity.

The fundamental functioning of the XGBoost algorithm lies in its iterative training process, wherein weak prediction models are sequentially incorporated into the en-semble while concurrently minimizing the overall prediction error. Each subsequent model concentrates on capturing the residual errors of its predecessors, progressively refining the predictive capabilities of XGBoost. By skillfully combining gradient boosting and regularization techniques, XGBoost adeptly balances model complexity and generalization prowess, culminating in superior performance.

Within the context of IoT botnet classification, XGBoost affords several notable advantages. It effortlessly navigates the challenges posed by high-dimensional and heterogeneous data frequently encountered in IoT environments, thereby facilitating the incorporation of diverse features and patterns intrinsic to botnet activities. Additionally, XGBoost excels at capturing intricate relationships and interactions among these features, heightening its ability to accurately classify IoT botnets. Moreover, XGBoost seamlessly integrates regularization techniques that effectively mitigate the perils of overfitting, enhancing its robustness and generalization capabilities.

Nevertheless, XGBoost encounters challenges in terms of interpretability. As an en-semble of decision trees, elucidating the individual contributions of each tree within the XGBoost model can prove arduous. Nonetheless, techniques centered around feature importance analysis can be employed to gain insights into the relative significance of features in the classification process.

Extreme Gradient Boosting (XGBoost) emerges as a potent and commendable algorithm within the domain of IoT botnet classification. Through its ensemble learning approach, amalgamated with gradient boosting and regularization techniques, XGBoost adeptly captures intricate relationships and deftly handles high-dimensional data. Notwithstanding potential challenges pertaining to interpretability, XGBoost's resolute performance and accurate classification of IoT botnets underscore its indispensable role in the cybersecurity domain.

4. Performance Evaluation

In assessing the algorithmic models described earlier, diverse techniques were em-ployed to gauge the precision of the outcomes and derive comprehensive findings for each model. This study involved several fundamental concepts, such as TP, TN, FP, and FN. TP signifies the count of true positives that have been accurately determined, while TN is the number of true negatives that have been correctly identified. FP represents the actual number of positive cases that have been erroneously classified as negative, whereas FN denotes the count of negative cases that have been mistakenly classified as positive.

a) Precision (PRE)

Precision is a performance metric utilized to assess a model's effectiveness by determining the proportion of accurately identified positive instances. This measure can be mathematically calculated through the use of a formula, which takes into account the number of true positives and false positives. Specifically, precision is computed by dividing the number of true positives by the sum of true positives and false positives.

$$PRE = TP/(TP + FP) \tag{3}$$

b) Accuracy (ACC)

Accuracy is a performance metric utilized to gauge a model's efficacy by determining the proportion of correct predictions out of the total number of predictions. This measure can be mathematically calculated through the use of a formula, which considers the number of true positives and true negatives. Specifically, accuracy is computed by dividing the sum of true positives and true negatives by the total number of predictions.

$$ACC = (\mathrm{TP} + \mathrm{TN})/(\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}) \tag{4}$$

c) Recall Score (RE)

The recall score is a performance metric utilized to evaluate a model's efficacy in correctly identifying actual positive instances. This measure can be mathematically calculated through the use of a formula, which incorporates the number of true positives and false negatives. Specifically, the recall score is computed by dividing the number of true positives by the sum of true positives and false negatives.

$$RE = TP/(TP + FN) \tag{5}$$

d) F1 Score for Binary class

The F1 score is a performance metric that is obtained by averaging both the accuracy and recall scores. This measure is widely used in assessing the overall effectiveness of a model since it provides a comprehensive view of false positives and false negatives. Specifically, the F1 score is calculated as the harmonic mean of precision and recall. The formula for computing the F1 score takes into account both the number of true positives, false positives, and false negatives, thereby providing a more balanced evaluation of a model's performance.

$$F1 - \text{ score } = 2 \times \frac{PRE \times RE}{PRE + RE} \tag{6}$$

e) F1 Score for Multi-class

The F1 score [16] is a widely recognized and commonly employed metric for evaluating the performance of multi-class classification algorithms. It serves as a comprehensive measure that takes into account both precision and recall, thereby enabling a balanced assessment of the classifier's overall efficacy.

Within the context of multi-class classification in the IoT-botnet problem, the F1 score offers valuable insights into the algorithms' capability to accurately classify instances across diverse classes. By considering the occurrence of false positives (misclassified instances) and false negatives (missed instances) for each class, the F1 score provides a robust evaluation of the algorithm's performance.

The calculation of the F1 score involves determining the harmonic mean of precision and recall for each class. This approach effectively evaluates the equilibrium between correctly identifying positive instances (precision) and capturing all positive instances (recall). A higher F1 score signifies a superior balance between precision and recall, thereby demonstrating the classifier's competence in accurately identifying instances across all classes.

To summarize, the F1 score assumes a pivotal role as a critical metric in assessing the performance of multi-class classification algorithms in the IoT-botnet problem. It facilitates a comprehensive evaluation by considering precision and recall for each class, thereby enabling a well-rounded assessment of the classifier's overall effectiveness in accurately classifying instances across multiple classes.

IV. Results and Discussion

1. Binary Classification

Table 5 presents a comprehensive analysis of the evaluation results obtained from binary classification in the IoT-botnet problem. Four distinct machine learning algorithms, namely Decision Tree (DT), Random Forest (RF), k-Nearest Neighbor (KNN), and Extreme Gradient Boosting (XGB), were employed to discern their performance using various metrics.

Table 5. Some metrics of binary classification

Evaluation	DT	RF	KNN	XGB
ACC	99.9%	89.5%	99.6%	99.8%
PRE	100.0%	88.0%	100.0%	100%
RE	100.0%	86.0%	99.0%	100%
F1	100.0%	87.0%	100.0%	100%
Time (s)	5.9	40.2	58.2	99.4

In terms of accuracy (ACC), the Decision Tree algorithm exhibited unparalleled precision, achieving an exceptional accuracy rate of 99.9%. Close on its heels, Extreme Gradient Boosting demonstrated an impressive accuracy of 99.8%. k-Nearest Neighbor showcased commendable accuracy at 99.6%, while Random Forest achieved a respectable accuracy of 89.5%.

Precision (PRE), which measures the ability to accurately classify positive instances among all instances predicted as positive, yielded noteworthy outcomes. Both Decision Tree and Extreme Gradient Boosting achieved flawless precision scores of 100.0%. k-Nearest Neighbor also showcased excellent precision, matching the perfect score of 100.0%. Although slightly lower, Random Forest performed commendably with a precision score of 88.0%.

The evaluation of Recall (RE), which quantifies the proportion of correctly classified positive instances among all actual positive instances, revealed exemplary results. Both Decision Tree and Extreme Gradient Boosting excelled with perfect recall scores of 100.0%. k-Nearest Neighbor demonstrated a commendable recall rate of 99.0%, while Random Forest achieved a respectable recall of 86.0%.

The F1 score, representing the harmonic mean of precision and recall, provided an overall assessment of the classifiers' performance. Both Decision Tree and Extreme Gradient Boosting achieved perfect F1 scores of 100.0%. k-Nearest Neighbor showcased excellent performance with an F1 score of 100.0%, while Random Forest achieved a respectable score of 87.0%.

In terms of computational time, the Decision Tree algorithm emerged as the most efficient, requiring a mere 5.9 seconds for processing. Random Forest followed with a processing time of 40.2 seconds, while k-Nearest Neighbor demanded 58.2 seconds. Extreme Gradient Boosting exhibited the longest processing time, clocking in at 99.4 seconds.

2. Multi classification

Table 6 illustrates the comprehensive evaluation and analysis results pertaining to multi-classification in the context of the IoT-botnet problem. The study involved the assessment of four distinct machine learning algorithms, namely Decision Tree (DT), Random Forest (RF), k-Nearest Neighbor (KNN), and Extreme Gradient Boosting (XGB), utilizing a range of performance metrics.

Table 6. Some metrics of multi-classification

Evaluation	DT	RF	KNN	XGB
ACC	99.9%	78.2%	99.8%	99.9%
PRE	99.9%	59.6%	99.7%	100.0%
RE	99.7%	46.1%	98.7%	99.8%
F1	99.6%	49.8%	99.2%	99.9%
Time (s)	11.5	206.1	85.2	956.2

The primary metric of accuracy (ACC) reveals the remarkable performance achieved by both the Decision Tree and Extreme Gradient Boosting algorithms, attaining high accuracy rates of 99.9%. Following closely, k-Nearest Neighbor demonstrated commendable accuracy at 99.8%, while Random Forest exhibited a slightly diminished accuracy of 78.2%.

Precision (PRE), which assesses the ability to correctly classify instances within each class, underscores the exceptional precision scores attained by the Decision Tree (99.9%) and Extreme Gradient Boosting (100.0%) algorithms. k-Nearest Neighbor demonstrated a commendable precision of 99.7%, while Random Forest displayed a relatively lower precision of 59.6%.

The metric of recall (RE), which reflects the sensitivity to correctly classify instances within each class, showcased outstanding results for the Decision Tree (99.7%) and Extreme Gradient Boosting (99.8%) algorithms. k-Nearest Neighbor exhibited a recall rate of 98.7%, while Random Forest achieved a recall of 46.1%, indicating a relatively lower performance in this aspect.

The F1 score, serving as a comprehensive evaluation of both precision and recall, further highlights the remarkable performance of the Decision Tree (99.6%) and Extreme Gradient Boosting (99.9%) algorithms. k-Nearest Neighbor achieved a respectable F1 score of 99.2%, while Random Forest yielded a comparatively lower score of 49.8%.

In terms of computational time, the Decision Tree algorithm demonstrated the shortest processing time of 11.5 seconds, showcasing its efficiency. k-Nearest Neighbor required 85.2 seconds, while Random Forest consumed a longer processing time of 206.1 seconds. On the other hand, the Extreme Gradient Boosting algorithm exhibited the highest computational demand, with a processing time of 956.2 seconds.

The findings of this evaluation highlight the superior performance of the Decision Tree and Extreme Gradient Boosting algorithms across multiple metrics, including accuracy, precision, recall, and F1 score. Additionally, the Decision Tree algorithm stands out for its computational efficiency. However, it is important to consider the specific requirements and priorities of the application when selecting the most suitable algorithm, taking into account a balanced consideration of accuracy, precision, recall, and computational efficiency. The results of this research offer valuable insights into the performance characteristics of various machine learning algorithms for multi-classification in IoT-botnet problems.

3. Models evaluation

In order to ascertain the suitability of a given model for a particular problem, performance evaluation techniques are utilized. A range of methodologies may be employed, including but not limited to accuracy, precision, recall, and F1-score measurements. These methods serve as reliable indicators of a model's ability to effectively address the original problem at hand.

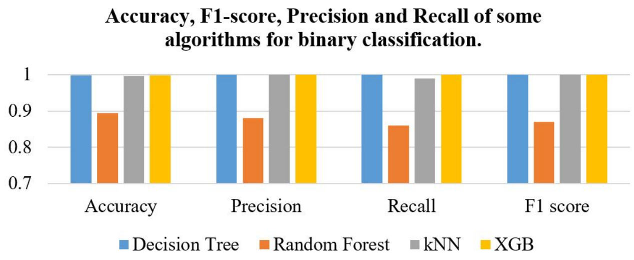


Figure 3: Accuracy, F1-score, Precision and Recall of some algorithms for binary classification

The performance evaluation of the DT, RF, KNN, and XGB algorithms for binary classification is presented in Figure 3, providing a comprehensive overview of the accuracy, precision, recall, and F1-score metrics. The results distinctly demonstrate the outstanding performance exhibited by both the DT and XGB algorithms, surpassing their respective counterparts. Additionally, in terms of training time, the Decision Tree algorithm emerges as the more efficient option, thereby delivering superior overall performance.

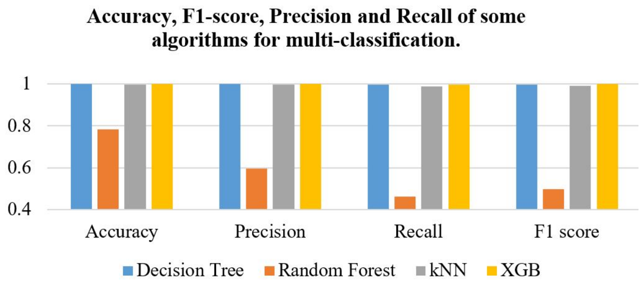


Figure 4: Accuracy, F1-score, Precision and Recall of some algorithms for multi-classification

The multiclass problem was evaluated to assess the performance of four distinct algorithms, namely Decision Tree, Random Forest, KNN, and XGB, as depicted in Figure 4. The evaluation encompassed the comprehensive analysis of accuracy, precision, recall, and F1-score metrics. Notably, the results demonstrate that the Decision Tree algorithm surpasses the other algorithms in terms of these performance metrics, showcasing superior performance.

Figure 5 presents the training time of the algorithms for the binary classification problem. The findings reveal that the XGB algorithm exhibits significantly longer training time compared to the other algorithms. Conversely, the decision tree algorithm emerges as the most efficient in terms of training time, outperforming all other algorithms in this regard.

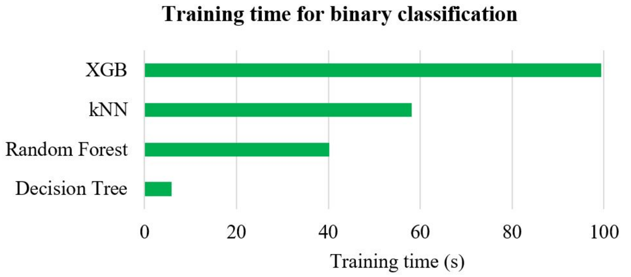


Figure 5: Training time for binary classification

In the multi-classification problem, the training time of the algorithms is presented in figure 6. The results illustrate that the XGB algorithm takes significantly longer to train compared to the other algorithms, with a training time almost 100 times that of the decision tree algorithm. On the other hand, the decision tree algorithm exhibits the best performance in terms of training time.

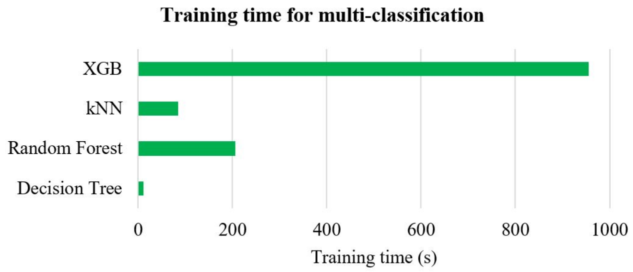


Figure 6: Training time for multi-classification

V. Conclusions

In conclusion, this scholarly research paper has undertaken the critical task of detecting and categorizing IoT botnets using machine learning algorithms. The study has placed considerable emphasis on the thorough analysis and manipulation of IoT botnet data, with a specific focus on the highly regarded IoT-23 dataset. By deploying widely recognized and extensively utilized machine learning algorithms such as Decision Trees (DT), k-Nearest Neighbors (KNN), Random Forests (RF), and eXtreme Gradient Boosting (XGBoost), the objective was to effectively classify and detect botnets within the confines of the IoT-23 dataset. The application of these algorithms has aimed to advance our understanding of their performance and effectiveness in the realm of IoT botnet detection and classification tasks.

Through a comparative analysis of the outcomes derived from these diverse algorithms, valuable insights have been acquired, shedding light on their respective strengths and limitations. Such insights equip researchers and practitioners with the knowledge required to make well-informed decisions when choosing the most appropriate algorithm for successful IoT botnet detection and classification. This research contributes significantly to the field by providing a comprehensive evaluation of the performance of these machine learning algorithms, thereby facilitating the development of more resilient and efficient detection and classification methodologies.

By addressing the pivotal challenge of IoT botnet detection and classification, this study has implications that extend to the realm of enhancing cybersecurity measures within IoT ecosystems. The findings and insights gleaned from this research have the potential to inform the development of proactive measures aimed at preventing and mitigating the detrimental effects of IoT botnet attacks. It is anticipated that this work will inspire further research endeavors and foster collaboration within the field, ultimately leading to the advancement of more secure IoT systems and the safeguarding of critical infrastructures.

References

[1] Williams, P., Dutta, I.K., Daoud,. H, and Bayoumi, M.: A survey on security in internet of things with a focus on the impact of emerging technologies. Elsevier. (2022).

[2] Khan, W.Z., Rehman, M.H., Zangoti, H.M., Afzal, M.K., Armi, N., and Salah, K.: Industrial internet of things: Recent advances, enabling technologies and open challenges. Elsevier. (2020).

[3] Vitorino, J., Andrade, R., Praça, I., Sousa, O., and Maia, E.: A Comparative Analysis of Machine Learning Techniques for IoT Intrusion Detection. Foundations and Practice of Security (pp.191-207).Springer.(2022).

[4] Haq, N.F., Onik, A.R., Hridoy, M.A.K, Rafni, M., Shah, F.M., and Farid, D.M.: Application of Machine Learning Approaches inIntrusion Detection System: A Survey. Article Published in International Journal of Advanced Research in Artificial Intelligence(IJARAI), Volume 4 Issue 3. (2015).

[5] Hajji, J., Khalily, M., Moustafa, N., and Nelms, T. IoT-23: A Dataset for IoT Network Traffic Analysis. Springer. (2019).

[6] Rahim, A., Razzaque, M.A., Hasan, R., and Hossain, M.F. Effective IoT Network Security through Feature Selection and Machine Learning Techniques. IEEE. (2020).

[7] Islam, S.M.Z, Bhuiyan, M.Z.H, and Hasan, R. Fusion of Machine Learning Models for Intru-sion Detection in IoT Networks using the IoT-23 Dataset. IEEE. (2020)

[8] Li, Y., Qiu, L., Chen, Y., and Chen, Y. Ensemble-based Intrusion Detection System for IoT Networks using the IoT-23 Dataset. IEEE. (2020)

[9] P. H. Do, T. D. Dinh, D. T. Le, V. D. Pham, L. Myrova and R. Kirichek, ”An Efficient Fea-ture Extraction Method for Attack Classification in IoT Networks,” 2021 13th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)

[10] Alotaibi, F., Al-Qaness, M.A., Abunadi, A., and Al-ghazzawi, M.A. A Deep Learning Ap-proach for Intrusion Detection in IoT Networks using the IoT-23 Dataset. IEEE. (2020).

[11] Li, J., Hu, C., Yang, K., Zhang, X., and Lu, J. An IoT-23 based IoT Intrusion Detection Sys-tem using Deep Learning. IEEE. (2020).

[12] Abdallah, A., Khalil, I., Al-Emadi, N., Almohaimeed, A., and Kim, H. Real-Time IoT Botnet Detection Using Deep Learning on IoT-23 Dataset. IEEE. (2020)

[13] Kiani, A.T., Abbas, R.A., Abbasi, A.Z., and Khan,

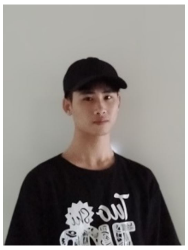
M.K. Deep Learning-based Anomaly De-tection for IoT Networks using the IoT-23 Dataset. IEEE. (2020)

[14] Rasool, S., Saeed, S., Farooq, F., and Madani, A. A Comparative Study of Transfer Learning Approaches for IoT Malware Detection Using IoT-23 Dataset. IEEE. (2021).

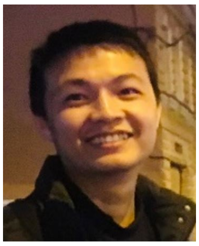
[15] Sebastian Garcia, Agustin Parmisano, and Maria Jose Erquiaga. (2020). IoT-23: A labeled dataset with malicious and benign IoT network traffic (Version 1.0.0) [Data set]. Zenodo. http://doi.org/10.5281/zenodo.4743746

[16] Stoian, N.A. Machine Learning for Anomaly Detection in IoT Networks : Malware analysis on the IoT-23 data set. EEMCS: Electrical Engineering, Mathematics and Computer Science. (2020)

Pham Van Quan is currently a senior student majoring in Data Science and Artificial Intelligence in Dong A University. His research interests include DS, ML, AI and its application in different fields like Finance, Network and Natural Language Processing.



Ngo Van uc became a student at Dong A University in 2020, majoring in Artificial Intelligence and Data Science. His research interests include machine learning, deep learning, data science, artificial intelligence, image processing, and their applications



Do Phuc Hao received his MS degree in Computer science from the University of Danang - University of Science and Technology in 2017. He is currently a Ph.D. student in the Department of Communication Networks and Data Transmission at the Bonch-Bruevich Saint- Petersburg State University of Telecommunications, Russia. His research interests include Artificial Intelligence, Machine Learning and its application in different fields like network, blockchain.

Nguyen Nang Hung Van received his Ph.D. degree in Computer Science from The University of Danang, Vietnam, in 2021. He is currently a lecturer at The University of Danang - University of Science and Technology. His research interests include Artificial Intelligence, Machine Learning, Geometric Algebra, Computer Networking.