universität freiburg

Embedded Systems Entrepreneurship (2ES)

Vishal Sivakumar (5253589), Sakshi Kulkarni (5586550)

Intrusion Detection in IOT Networks

Date: August 6, 2023
Supervisors: Prof. Dr. Oliver Amft, Prof. Dr. Axel Sikora

Abstract

The development of smart devices and the Internet of Things (IoT) has significantly impacted various aspects of our lives. For manufacturing companies, IoT technologies offer real-time monitoring of machines, product quality, and environmental variables, enabling managers to make informed decisions and reduce risks effectively. However, the widespread use of IoT devices also poses security and safety challenges, necessitating the detection of intrusions in IoT networks.

To address this issue, our report proposes the application of Machine Learning techniques to detect intrusions in IoT networks. We conducted experiments using the IoT-23 dataset to compare the performance of various models. By identifying the best algorithm with high efficiency and accuracy, we aim to enhance IoT network security and prevent potential damages or losses effectively.

Contents

1 Introduction

The Internet of Things (IoT) has emerged as a groundbreaking innovation in the global information industry, following the Internet. It comprises a smart network enabling devices to communicate and exchange data through the internet. IoT allows for tracking, monitoring, locating, identifying, and managing various items in our daily lives. As the number of IoT devices increases in different sectors like Smart Healthcare, Smart Transportation, Smart Governance, Smart Agriculture, etc., it has become a prominent research area in computer science.
With the convenience IoT brings, human behavior has changed, especially among younger generations who heavily rely on IoT devices like smart bulbs, ovens, refrigerators, temperature sensors, and smoke detectors. However, this growing reliance has raised privacy and security concerns. As all devices are connected to the internet and each other, it creates more potential ways for attackers to access information. The collection of personal data by connected devices makes users vulnerable to information theft and even control of their smart devices as shown in figure 1. This hampers the progress of IoT technology and infrastructure development, necessitating a focus on ensuring the security and privacy of these interconnected devices.

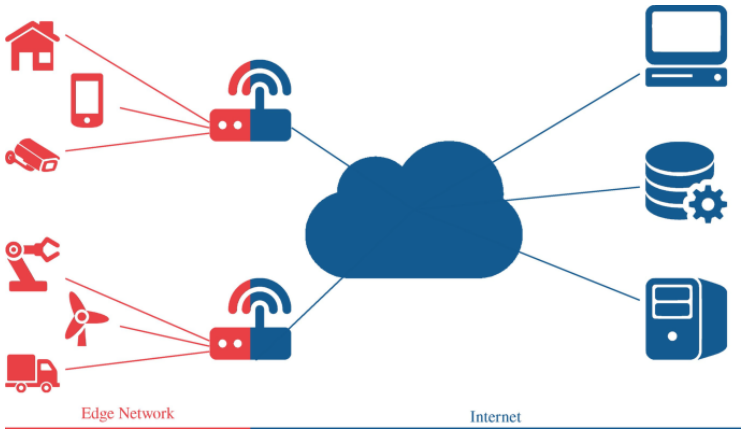

Figure 1: IoT Networks

To address these challenges, we propose the use of Machine Learning (ML) algorithms, which have shown promise in distinguishing between regular and malicious network behavior. By analyzing data from trained algorithms, we can detect abnormal activities in the network, thereby preventing unauthorized access. In our study, we aim to utilize lightweight ML methods to improve the accuracy of detecting malicious nodes. The central unit captures IoT traffic data and feeds it to selected trained ML models. Multiple ML models will be

tested to cater to the distinct needs of different users or groups.
The sheer volume and diversity of data within IoT networks make enhancing security and meeting various requirements (cost-effectiveness, reliability, etc.) challenging. Striking a balance is crucial, as enhancing one feature may affect the performance of others. For instance, increasing security checks and protocols may lead to higher costs and latency, making some applications impractical for certain users. Furthermore, the surge in connected devices increases the potential for attackers to exploit weak links, like devices lacking proper security features such as firewalls and anti-virus protection. Given the resource-constrained nature of IoT devices, it is crucial to detect intrusions with less complexity and time. ML techniques can help mitigate complexity as these models learn from trained data.

Addressing the privacy and security concerns of IoT motivates our research in developing a framework for automatic IoT sensors attack and intrusion detection. In this report, we propose to employ ML algorithms such as Support Vector Machines, Random Forest, and Naive Bayes for intrusion detection. We will evaluate the performance of these models using the IoT-23 dataset to identify the most suitable algorithm based on accuracy and time cost. The results of this study will contribute to enhancing the security and privacy of IoT networks.

2 Literature Review

An IoT botnet as shown in figure 2, is a specific type of botnet where the compromised devices are IoT devices, sharing similarities in structure and behavior with computer botnets. When a vulnerable IoT device gets compromised, it becomes a part of a larger community of compromised devices, referred to as a botnet, under the command of a malicious actor known as the botmaster. The botmaster gains remote access and control over the compromised devices through the Internet, without the knowledge or consent of the device's legitimate owner, by utilizing a Command & Control (C& C) server.

Botnets [3] have been responsible for carrying out exceptionally large-scale Distributed Denial of Service (DDoS) attacks. For instance in [6], in 2016, Brian Krebs, a journalist, experienced a record-breaking attack of 620 Gbps directed at his blog KrebsOnSecurity.com, causing the site to go offline (Krebs, 2016). Similarly, the French hosting provider OVH faced an attack from the same botnet, reaching an astounding 1 Tbps and involving over 140,000 compromised cameras and DVRs (Pritchard, 2018 in [7]). The infamous Mirai botnet targeted the domain name system provider Dyn, using approximately 100,000 IoT devices and causing major websites like CNN, Netflix, and Paypal to become inoperative for hours, with peak traffic reaching 1.2 Tbps (Weisman, 2019 in [8]). These attacks resulted in significant losses in revenue, customers, and trust for the affected companies.

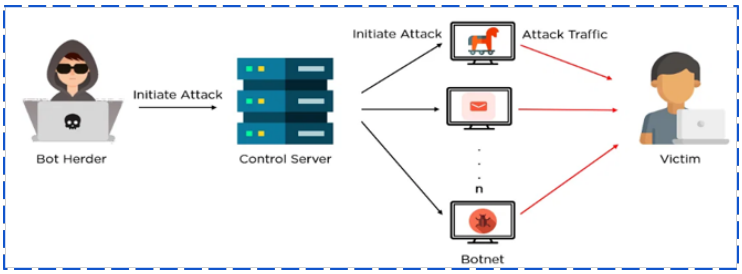

Figure 2: Working of a Botnet

As a response to these threats, intrusion detection systems play a vital role in network security, focusing on identifying security threats when preventive measures are inadequate to address vulnerabilities. These systems enable timely identification and mitigation of attacks, enabling businesses to protect their IoT infrastructure and sensitive data from unauthorized access.

In 2018 Machine Learning DDoS Detection for Consumer Internet of Things Devices[4], provides an overview of the research on automatically detecting distributed denial of service (DDoS) attacks in Internet of Things (IoT) networks using machine learning algorithms. The increasing number of IoT devices con-

necting to the internet has led to security vulnerabilities and botnet attacks. The paper proposes a machine learning-based approach that leverages IoT-specific network behaviors for accurate DDoS detection.

The paper presents a four-step anomaly detection pipeline that involves traffic capture, grouping packets by device and time, feature extraction, and binary classification. It explores two classes of features: stateless features derived from individual packet characteristics and stateful features capturing network traffic evolution over time.

Five machine learning algorithms, namely K-nearest neighbors, Support vector machine with a linear kernel, decision tree, Random forest, and Neural Network, were tested for classifying normal and DDoS attack traffic. The results showed high accuracy for all classifiers, indicating the effectiveness of the proposed approach.

The paper discusses the importance of lightweight and protocol-agnostic features, suitable for deployment on network middleboxes, such as routers and firewalls. It emphasizes the significance of distinguishing IoT-specific network behaviors to identify IoT botnets at the local network level.

The threat model assumes a scenario where an on-path device can observe traffic between IoT devices and the internet. The goal is to detect and prevent DDoS attacks originating from devices within the smart home LAN. The time range of DoS attacks is assumed to be approximately 1.5 minutes to avoid detection.

Future work is suggested to replicate the results with additional IoT devices and real DDoS attack traffic. The study also encourages experimenting with more features and complex machine learning techniques to enhance anomaly detection and develop network protection strategies for IoT devices.

Overall, the paper demonstrates the potential of machine learning-based anomaly detection to protect IoT networks from DDoS attacks and emphasizes the need for specialized features to address the unique characteristics of IoT traffic.

3 Methods

This study proposes an intrusion detection system model for IoT security. The model aims to efficiently detect intrusions in IoT network traffic using Machine Learning (ML) algorithms.

In the proposed model, we consider a recent dataset for detection that is IoT-23 [2]. IoT-23 is a new dataset of network traffic from Internet of Things (IoT) devices. It has 20 malware captures executed in IoT devices, and 3 captures for benign IoT devices traffic. It was first published in January 2020, with captures ranging from 2018 to 2019. This IoT network traffic was captured in the Stratosphere Laboratory, AIC group, FEL, CTU University, Czech Republic. Its goal is to offer a large dataset of real and labeled IoT malware infections and IoT benign traffic for researchers to develop machine learning algorithms. This dataset and its research is funded by Avast Software, Prague [**?**]. The compute unit runs multiple ML models to assess their performance and cost.

The ML models' performance and cost metrics are analyzed to enable users or the system to select the most suitable model for intrusion detection based on factors like accuracy, precision, and f1-score. This flexibility ensures that different users with unique requirements for an intrusion detection system can find an appropriate solution.

The ML algorithms used in this study include Naive Bayesian, Support Vector Machine (SVM), Random Forest Classification, Multilayer Perceptron Classifier and Naive Bayesian Ada-Boost Algorithm. Lets study these models in detail below.

3.1 Naive Bayesian

Naive Bayes is a probabilistic machine learning algorithm used for classification tasks[9]. It is based on Bayes' theorem, which is a fundamental concept in probability theory. The algorithm is particularly well-suited for text and categorical data but can be applied to other types of data as well.

The basic idea behind Naive Bayes is to determine the probability of an observation (e.g., a data point) belonging to a particular class, given its feature values. It assumes that the features are conditionally independent of each other, meaning that the presence or absence of one feature does not affect the presence or absence of other features. This is where the "naive" assumption comes into play.

The Naive Bayes algorithm consists of three main steps:

1. Data Preparation: The dataset is prepared with labeled examples, where each observation has a set of features and a corresponding class label.

2. Training: During the training phase, Naive Bayes estimates the probabilities of each class and the conditional probabilities of feature values given each

class. These probabilities are calculated based on the frequencies of features in the training data.

3. Classification: When a new observation needs to be classified, Naive Bayes calculates the probability of the observation belonging to each class using Bayes' theorem. The class with the highest probability is selected as the predicted class for the observation.

Naive Bayes is computationally efficient and can handle a large number of features, making it particularly useful for high-dimensional data. It is widely used in various applications such as spam email detection, sentiment analysis, document categorization, and more.

However, it's essential to note that the "naive" assumption of feature independence may not always hold true in real-world data, and this can impact the algorithm's performance in certain situations. Despite this limitation, Naive Bayes often performs surprisingly well, especially when the data follows the conditional independence assumption or in scenarios where computational efficiency is crucial.

3.2 Support Vector Machine

Support Vector Machine (SVM) is a powerful supervised learning algorithm commonly used for classification and regression tasks[10]. In the context of the proposed intrusion detection system for IoT security, SVM serves as one of the fundamental machine learning techniques to distinguish between normal and malicious network traffic.

The process begins with data preparation, where a labeled dataset is assembled, consisting of observations (e.g., network traffic data) with corresponding class labels denoting whether they are normal or anomalies.

During the training phase, SVM's objective is to discover the optimal hyperplane that maximizes the margin between the two classes as shown in figure 3[10]. The margin represents the distance between the hyperplane and the closest data points of each class. By optimizing this margin, SVM creates an effective decision boundary that effectively separates the classes.

To handle scenarios where the data is not linearly separable in the original feature space, SVM employs the "kernel trick." This technique transforms the data into a higher-dimensional space where it becomes linearly separable. Various kernel functions, such as linear, polynomial, radial basis function (RBF), and sigmoid kernels, can be utilized for this purpose[10].

When a new observation (e.g., incoming network traffic) requires classification, SVM places it in the appropriate class based on its position relative to the decision boundary. Data points on one side of the boundary are classified as normal, while those on the other side are classified as anomalies.

By learning the optimal hyperplane during training, SVM becomes proficient at detecting intrusions during the classification process. It identifies potential

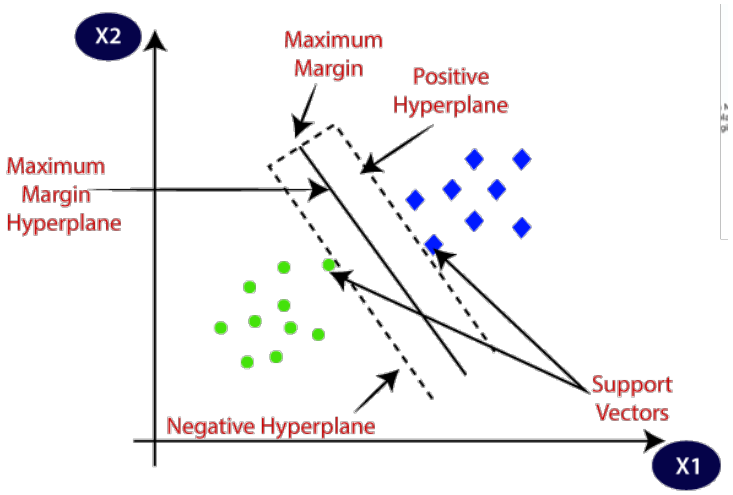

Figure 3: SVM representation

intrusions by determining whether a new observation lies within the region corresponding to anomalous network behavior.

SVM boasts several advantages, including its ability to handle high-dimensional data effectively, its robustness against overfitting, and its proficiency in dealing with complex data distributions. However, it's crucial to carefully tune SVM's hyperparameters, as its performance can be sensitive to their settings[10]. Additionally, training SVM on large datasets may lead to computational challenges due to its complexity.

In conclusion, Support Vector Machine (SVM) plays a pivotal role in the proposed intrusion detection system, accurately classifying IoT network traffic as either normal or anomalous. By finding the optimal hyperplane to separate the two classes, SVM contributes to efficient and reliable intrusion detection in IoT security scenarios.

3.3 Random Forest Classification

Random Forest is an ensemble learning algorithm used for classification tasks, including intrusion detection in the proposed IoT security system. It is a powerful and versatile algorithm known for its ability to handle complex and high-dimensional datasets effectively[11].

In a Random Forest classification, multiple decision trees are built during the training phase as shown in figure 4. Each decision tree is constructed using a random subset of the training data (sampling with replacement) and a random subset of features[11]. This randomness introduces diversity among the trees, reducing the risk of overfitting and enhancing the model's generalization capability.

When a new observation needs to be classified, each decision tree in the

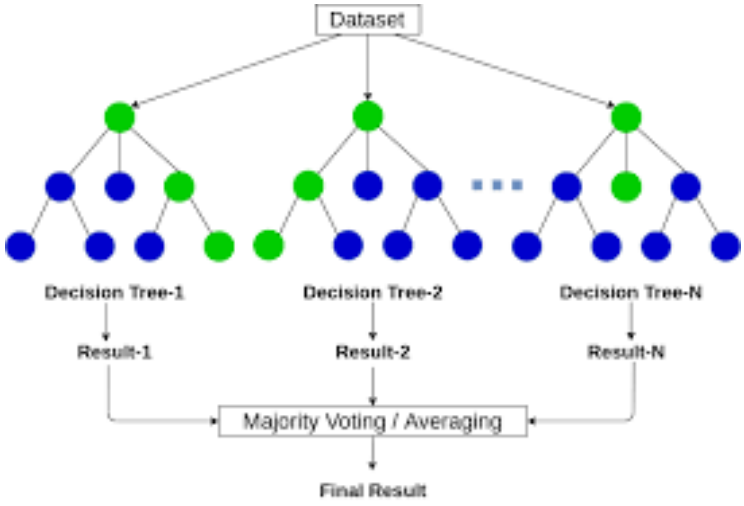

Figure 4: Random Forest Classification

Random Forest independently makes a prediction. For binary classification, the final prediction is determined by the majority vote among the decision trees. The observation is classified based on the class with the most votes.

Random Forest is capable of detecting intrusions by leveraging the combined strength of multiple decision trees. Since each tree is constructed with randomness, the ensemble can effectively identify patterns in the data and detect intrusions even in complex and noisy datasets.

Random Forest offers several advantages, such as handling high-dimensional data, capturing feature importance, and avoiding overfitting. It can also handle missing values and outliers effectively, making it a suitable choice for various real-world scenarios[11].

The algorithm's training process involves building multiple trees and aggregating their results, making it computationally efficient for large datasets. The model's robustness and accuracy make it a popular choice in various machine learning applications, including intrusion detection in IoT security.

3.4 Multilayer Perceptron Classifier

Multilayer Perceptron (MLP) is a type of Artificial Neural Network (ANN) frequently used for classification tasks, including intrusion detection in the proposed IoT security system. It is known for its versatility and ability to handle complex relationships in data[12].

An MLP consists of multiple layers of interconnected nodes, or neurons as shown in figure 5[12]. The first layer is the input layer, which receives the feature values of the observations (e.g., network traffic data). The output layer produces the final classification result, while the hidden layers assist the net-

work in learning complex representations of the data.

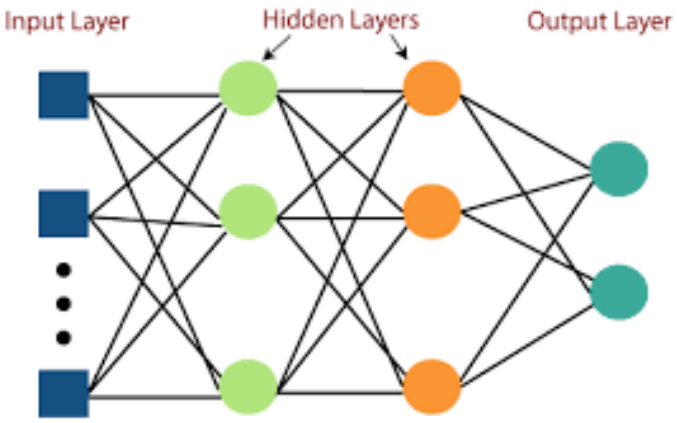

Figure 5: Multilayer Neural Network

During training, the input data flows forward through the network from the input layer to the output layer. Each neuron multiplies its input by weights and applies an activation function to introduce non-linearity. The output of each neuron becomes the input for the subsequent layer, continuing until the output layer is reached.
The network compares its predictions to the actual labels in the training data to calculate the prediction error. The back propagation algorithm adjusts the weights in each neuron's connections to minimize the error and improve the model's accuracy[12].
Once trained, the MLP can make predictions about whether incoming network traffic is normal or represents an intrusion. The output layer produces a probability score or a class label, indicating the likelihood of the observation being an intrusion.
MLPs are valuable for their ability to learn complex relationships and handle various types of data[12]. However, training an MLP may require more data and computational resources compared to simpler algorithms like Naive Bayes or SVM. Nevertheless, it offers high accuracy and robustness, making it a popular choice for intrusion detection and other machine learning tasks.

3.5 Naive Bayesian Ada-Boost Algorithm

The Naive Bayesian Ada-Boost Algorithm is an advanced machine learning approach that combines the principles of two powerful techniques: Naive Bayes and Ada-Boost. In this context, Ada-Boost, or Adaptive Boosting, is an ensemble learning algorithm that builds multiple weak classifiers and aggregates

their predictions to create a strong, accurate classifier[13]. On the other hand, Naive Bayes is a probabilistic classifier based on Bayes' theorem, known for its simplicity and effectiveness in various applications[9].

The Naive Bayesian Ada-Boost Algorithm operates as follows: Initially, the dataset is prepared with labeled examples, where each observation contains a set of features and a corresponding class label (e.g., normal or intrusion for network traffic data). Each observation in the dataset is assigned an initial weight, and all observations are given equal weight at the beginning[13].

The algorithm iterates through a process of building weak classifiers in successive rounds. During each round, the focus is on the most misclassified observations from the previous iteration. The algorithm adjusts the weights of misclassified observations, increasing them to emphasize the importance of difficult-to-classify instances and decreasing the weights of correctly classified observations.

For the classification step, a weak classifier, such as a Naive Bayes classifier, is trained using the weighted dataset. The weak classifier's performance is evaluated, and its weight in the final classification is determined based on its accuracy. More accurate classifiers are given higher weight in the final model, leading to a more robust and accurate ensemble classifier.

The predictions of the weak classifiers are then combined based on their weights to create a strong classifier. By assigning more weight to the predictions of the more accurate weak classifiers, the final model achieves a higher level of accuracy and robustness. When a new observation needs to be classified, the Naive Bayesian Ada-Boost Algorithm combines the predictions of all weak classifiers to make the final prediction. The observation is assigned to the class with the highest combined vote from the weak classifiers.

The Naive Bayesian Ada-Boost Algorithm is particularly valuable for complex and challenging classification tasks, making it an ideal choice for applications like intrusion detection in IoT security. By leveraging the strengths of both Naive Bayes and Ada-Boost, this algorithm can effectively handle diverse and intricate data patterns, leading to improved accuracy and performance in intrusion detection and other classification tasks.

4 Study

At the beginning of our project, our main objective was to find an appropriate IoT dataset that contains instances of malware and can be utilized for training machine learning algorithms. To achieve this, we extensively explored various datasets listed in the provided table.

Table 1: Data-sets for IoT anomaly-based Intrusion Detection System

Sr.no.	Dataset	Botnets	ML techniques
1.	N-Baiot	Mirai, BashLite	Deep Autoencoders, Local Outlier Factor, One-Class Support Vector Machines and Isolation Forest algorithms
2.	Doshi et al. (2018)	DDoS attacks using Mirai-derived IoT botnets	k-Nearest Neighbors, Support Vector Machines, Decision Tree, Random Forest and Artificial Neural Networks algorithms
3.	McDermott et al. (2018)	Mirai	Text recognition deep learning algorithm
4.	Shire et al. (2019)	IoT Malware Traffic	Convolutional Neural Networks

With fewer instances of malware and a smaller dataset size in these other datasets, the latest IoT23 dataset emerges as the preferred choice overall.
IoT-23 is a new dataset of network traffic from Internet of Things (IoT) devices. It has 20 malware captures executed in IoT devices, and 3 captures for benign IoT devices traffic. It was first published in January 2020, with captures ranging from 2018 to 2019. This IoT network traffic was captured in the Stratosphere Laboratory, AIC group, FEL, CTU University, Czech Republic. Its goal is to offer a large dataset of real and labeled IoT malware infections and IoT benign traffic for researchers to develop machine learning algorithms. This dataset and its research is funded by Avast Software, Prague.[2] This labeled dataset contains a diverse range of IoT network traffic, including both malicious and benign samples. It encompasses twenty-three distinct captures or scenarios, each representing various IoT network activities. Among these scenarios, twenty captures pertain to infected IoT devices, with each scenario being associated with the name of the executed malware sample. Additionally, there are three captures of real IoT devices' network traffic, specifying the names of the devices where the traffic was captured.

To create the malicious scenarios, the researchers executed specific malware samples on a Raspberry Pi, utilizing various protocols and actions to emulate real-world attacks. The dataset's Table 1 provides detailed characteristics of the IoT botnet scenarios, while Table 2 elaborates on the protocols found in each network traffic capture.

Both the malicious and benign scenarios were conducted in a controlled network environment with an unrestrained internet connection, accurately mimicking typical real-world IoT devices' behavior and interactions.

The primary objective of this dataset is to provide the community with two types of datasets: one containing malicious network traffic and the other exclusively composed of benign IoT traffic.

Afterward, we conducted an in-depth examination of the dataset, meticulously studying its features. Following that, we ventured into the implementation of several machine learning algorithms, such as Naive Bayesian, Support Vector Machine (SVM), Random Forest Classification, Multilayer Perceptron Classifier, and Naive Bayesian Ada-Boost Algorithm.

Initially, we performed data preprocessing, which involved removing irrelevant columns and limiting values to ranges suitable for analysis. The dataset was then split into training and testing data to evaluate the performance of various algorithms based on their respective training results. To assess the effectiveness of each algorithm in detecting malware or benign attacks, we constructed a confusion matrix and calculated metrics such as precision, recall, F1-score, and support. Furthermore, we measured the accuracy of each algorithm in identifying potential attacks.

In brief, the proposed intrusion detection system for IoT security utilizes machine learning algorithms, empowering users to select the best-fitted model based on performance metrics. Its ultimate goal is to strengthen IoT security by efficiently identifying and addressing network anomalies.

5 Results

We tested five machine learning algorithms to distinguish normal IoT packets from DoS attack packets:
1) Naive Bayesian Algorithm
2) Support Vector Machine (SVM)
3) Random Forest Classification (RF)
4) Multilayer Perceptron Classifier (MLP)
5) Naive Bayesian Ada-Boost Algorithm

We implemented these machine learning models using the Scikit-learn Python library [**?**], except for the Multilayer Perceptron Classifier, which was implemented using the Keras library [5].

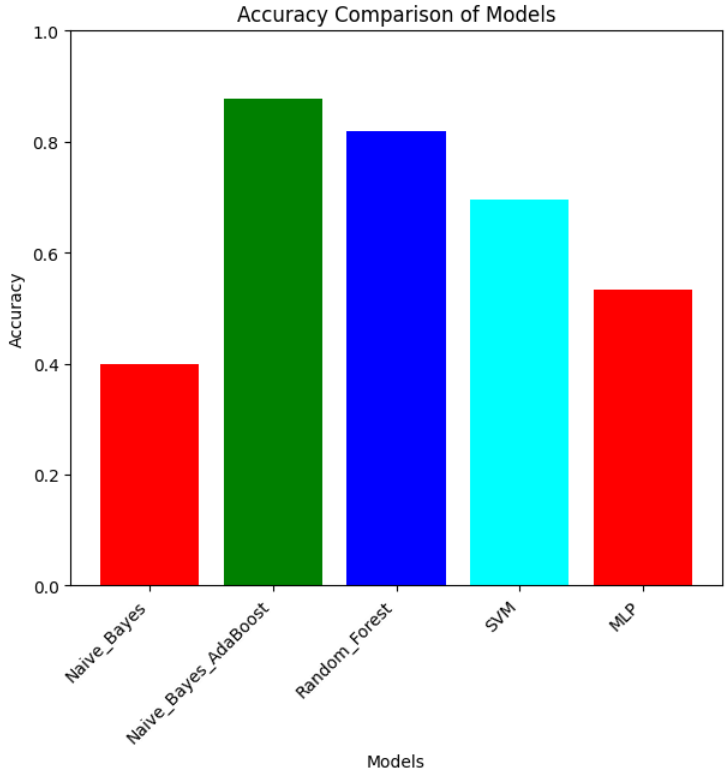

Figure 6: Comparison based on Accuracy of all methods

In Figure 6, the Accuracy comparison of the provided methods is depicted. When applied to the substantial IoT-23 dataset, these methods exhibit highly diverse performance.
Among these methods, Naive Bayesian Ada-Boost stands out with the highest accuracy of 91% compared to the rest.

Other performance metrics like precision, recall, F1-score, and support were also calculated as shown in Table 2

Table 2: Performance metrics

Parameters	Naive Bayesian Algorithm	Support Vector Machine (SVM)	Random Forest Classification	Naive Bayesian AdaBoost Algorithm
Precision(Attack)	0.79	0.67	1.00	1.00
Precision(Normal)	1.00	0.60	0.96	0.98
Recall(Attack)	0.98	1.00	1.00	1.00
Recall(Normal)	0.35	0.01	0.66	0.80
F1-score (Attack)	0.87	0.80	1.00	1.00
F1-score (Normal)	0.52	0.02	0.78	0.82
Support(Attack)	836	2877	773	834
Support(Normal)	33091	392	33131	33120

6 Discussion and Future work

The implementation of the proposed intrusion detection system has significant implications for businesses in the current world. By effectively detecting and responding to intrusions in IoT networks, businesses can enhance their security posture, protect critical data, and ensure smooth operations. One real-life example is a healthcare organization that relies on IoT devices for patient monitoring and medical data collection.
With the intrusion detection system in place, the organization can quickly identify and respond to any unauthorized access attempts or potential breaches of patient data, ensuring patient privacy and compliance with healthcare regulations. This not only protects the organization from potential security threats but also instills confidence among patients and healthcare professionals in the secure use of IoT technologies in the medical field.
This initial study shows that using basic classification algorithms and low-dimensional features can successfully differentiate between normal IoT device traffic and Malware attack. These findings encourage further investigation to assess IoT Malware detection in more realistic scenarios. To achieve this, we aim to reproduce the study's outcomes using additional datasets of normal traffic from various IoT devices and authentic attack traffic recorded from a real Malware attack.
Furthermore, there is a need to enhance the accuracy of the classification model, especially when dealing with large datasets. Scaling up the system to handle larger volumes of IoT network traffic while maintaining high precision is a crucial aspect of improving the overall effectiveness of intrusion detection. This involves optimizing the existing algorithms, exploring more sophisticated machine learning techniques, and potentially leveraging distributed computing to efficiently process and classify vast amounts of data in real time. The goal is to ensure that the intrusion detection system remains reliable, robust, and capable of accurately identifying intrusions amidst the ever-growing and diverse IoT network traffic.
We intend to conduct experiments involving additional features and explore more sophisticated machine-learning techniques beyond the ones presented in this research. We firmly believe that there is significant potential in utilizing deep learning approaches for intrusion detection in IoT networks. By delving into the realm of deep learning, we aim to unlock novel insights and further advancements in safeguarding IoT devices from intrusions and cyber threats. Our aspiration is that this study will serve as a catalyst, inspiring further endeavors in the development of specialized network protection methods tailored to the unique challenges posed by IoT devices.

7 Conclusion

In this study, we conducted tests using five different machine learning methods on the IoT23 dataset, which includes both malicious and benign IoT network traffic. Among these methods, Naive Based Ada-Boost and Random Forest exhibited relatively higher accuracy with the large dataset. These initial findings serve as a motivation for further research in the area of machine learning intrusion detection to enhance network security and protect against vulnerabilities posed by insecure IoT devices.

In addition to the preliminary results, our overarching goal is to develop more effective security models tailored for businesses using the IoT23 dataset. By refining machine learning intrusion detection techniques and harnessing the power of the dataset, we seek to create robust and advanced security measures that can safeguard businesses against potential threats and vulnerabilities arising from IoT devices. Our efforts are dedicated to bolstering network protection and ensuring a secure environment for IoT deployment in the business landscape.

References

[1] R. Doshi, N. Apthorpe and N. Feamster, "Machine Learning DDoS Detection for Consumer Internet of Things Devices," 2018 IEEE Security and Privacy Workshops (SPW), San Francisco, CA, USA, 2018, pp. 29-35, doi: 10.1109/SPW.2018.00013.

[2] Sebastian Garcia, Agustin Parmisano, & 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.

[3] A. Guerra-Manzanares, H. Galeana-Zapién, A. Flores-Rios, S. Villarreal-Reyes, R. Monroy-Beltrán, and C. M. Villanueva-Jiménez, "Medbiot: Generation of an IoT botnet dataset in a medium-sized IoT network," [Online]. Available: https://www.researchgate.net/profile/Alejandro-Guerra- Manzanares /publication/338765489/MedBIoTGenerationofanI oTBotnetDatasetinaM ediumsizedI oTN etwork/links/5e7d058292851caef 4a1ec74/MedBIoT-Generation-of-an-IoT-Botnet-Dataset-in-a-Medium-sized-IoT-Network.pdf

[4] Liang, Y. and Vankayalapati, N.V. (2021) YLIANG725/"Anomaly-detection-IOT23: A research project of anomaly detection on dataset IOT-23", GitHub. Available at: https://github.com/yliang725/Anomaly-Detection-IoT23 (Accessed: 06 July 2023).

[5] F. Chollet et al., "Keras," https://github.com/fchollet/keras, 2015.

[6] Krebs, B. (2016)." Krebs on security hit with record ddos". Retrieved from:https://krebsonsecurity.com/2016/09/krebsonsecurity-hit-with-record-ddos/.

[7] Pritchard, M. (2018). Ddos attack timeline: Time to take ddos seriously. Retrieved from: https://activereach.net/newsroom/blog/time-totake-ddos-seriously-a-recent-timeline-of-events/.

[8] Weisman, S. (2019). "Emerging threats - what is a distributed denial of service attack (ddos) and what can you do about them?" Retrieved from: https://us.norton.com/internetsecurityemerging-threats-what-is-a-ddosattack-30sectech-by-norton.html.

[9] F. -J. Yang, "An Implementation of Naive Bayes Classifier," 2018 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 2018, pp. 301-306,
doi: 10.1109/CSCI46756.2018.00065.

[10] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt and B. Scholkopf, "Support vector machines," in IEEE Intelligent Systems and their Applications, vol. 13, no. 4, pp. 18-28, July-Aug. 1998,doi: 10.1109/5254.708428.

[11] A. S. More and D. P. Rana, "Review of random forest classification techniques to resolve data imbalance," 2017 1st International Conference on Intelligent Systems and Information Management (ICISIM), Aurangabad, India, 2017, pp. 72-78, doi: 10.1109/ICISIM.2017.8122151.

[12] F. Amato, N. Mazzocca, F. Moscato and E. Vivenzio, "Multilayer Perceptron: An Intelligent Model for Classification and Intrusion Detection," 2017 31st International Conference on Advanced Information Networking and Applications Workshops (WAINA), Taipei, Taiwan, 2017, pp. 686-691,
doi: 10.1109/WAINA.2017.134.

[13] W. Li and Q. Li, "Using Naive Bayes with AdaBoost to Enhance Network Anomaly Intrusion Detection," 2010 Third International Conference on Intelligent Networks and Intelligent Systems, Shenyang, China, 2010, pp. 486-489, doi: 10.1109/ICINIS.2010.133.

List of Figures

List of Tables