BCSE353E – INFORMATION SECURITY ANALYSIS AND AUDIT

Prof. Sheikh Abdullah A (SCOPE)

BCSE353E – INFORMATION SECURITY ANALYSIS AND AUDIT

PROJECT REPORT ON

BOTNET DETECTION USING MACHINE LEARNING

By

Ashwath R - 21BCE1165 - School of Computer Science Engineering, VIT Chennai

Sriraam C - 21BCE5904 - School of Computer Science Engineering, VIT Chennai

Balaji R - 21BCE1307 - School of Computer Science Engineering, VIT Chennai

Kirtana A - 21BCE1321 - School of Computer Science Engineering, VIT Chennai

Abstract

The field of information and computer security is rapidly developing in today's world as the number of security risks is continuously being explored every day. The moment a new software or a product is launched in the market, a new exploit or vulnerability is exposed and is exploited by the attackers or malicious users for different motives. Many attacks are distributed in nature and carried out by botnets that cause widespread disruption of network activity by carrying out DDoS (Distributed Denial of Service) attacks, email spamming, click fraud, information and identity theft, virtual deceit and distributed resource usage for cryptocurrency mining.

Botnet detection is still an active area of research as no single technique is available that can detect the entire ecosystem of a botnet like Neris, Rbot, and Virut. They tend to have different configurations and heavily armoured by malware writers to evade detection systems by employing sophisticated evasion techniques.

This report provides a detailed overview of a botnet and its characteristics and the existing work that is done in the domain of botnet detection. The study aims to evaluate the preprocessing techniques like variance thresholding and one-hot encoding to clean the botnet dataset and feature selection technique like filter and wrapper to boost the machine learning model performance.

In this paper, performance of network dataset has been compared to predict the accuracy and anomalies on the network. The machine learning algorithms which have been used here is Random Forest, Extra Trees, Logistic Regression and Multinomial Naïve Bayes. Our experiments show, that our approach can compare the useful traffic and the junk traffic effectively and reaches the accuracy of 79.21%. Lastly, the optimal model is found by testing each model on the dataset of attacks and comparing its performance.

Keywords – Botnet Detection, Feature Selection, Variance Thresholding

Introduction

The internet is plagued with information theft and security risks. Information theft includes personal details stolen to conduct identity fraud, and debit and credit card credentials traded on the dark web to carry out illicit transactions. Some of the security risks include but are not limited to systems, servers, and networks compromised with malware, trojan horses, phishing, ad-wares, and viruses. While accessing resources like audio, video, and images and surfing the internet, users are targeted with unwanted ads, spam notification, and emails and denial of service. The attacks mentioned are carried out in a distributed manner for illegal purposes, monetary gains, to create biasedness among public opinion and harm the organization's reputation. Botnet comprises 32% of the attacks on the internet in the modern world. These nefarious activities are well organized and carried out by a hacker. A botnet is a network of malware compromised computers (called as bot or zombie) under the control of a hacker (also called as bot-herder or bot-master). A botherder controls the bots by using a Command-and-Control server (C&C). Identifying the vulnerable systems, propagating the malware, sending the command, and code updates and carrying out the attack are primarily controlled by the C&C server. A collective effort from the botnet attacks can result in Distributed Denial of Service (DDoS), phishing, spamming, spreading of malware, information theft, unwanted ads, generating virtual clicks and cryptocurrency mining. Prevention or detection of botnet attack is difficult because of its inherent nature of changing the attacks modus operandi. Many types of research have been done to effectively and successfully detect and block botnet attacks. The goal of this project is to propose a machine learning model to detect botnets using machine learning with better precision and reduce false positives by studying existing work done in the botnet detection area. The articles selected for this project include conference proceedings, articles and, published papers.

This project tries to answer the following questions:

1. How the dataset imbalance issue of botnet originated traffic can be handled?

2. Is there any machine learning model that can detect a range of botnet attacks?

Literature Review

In the realm of cybersecurity, the detection and mitigation of botnets pose significant challenges, necessitating the exploration of effective machine learning (ML) algorithms. Ensemble methods, such as random forest, have shown promise in handling large-scale problems, with "[2] A Random Forest Guide Tour" emphasizing its versatility and performance in diverse learning tasks. Another ensemble method, "[4] Extremely randomized trees," presents a unique approach with fully random split points, potentially offering advantages in botnet detection scenarios. Statistical models like logistic regression and Bayesian classification methods are scrutinized in "[3] Statistical comparison of logistic regression and different bayes classification methods for machine learning," evaluating their efficacy in identifying botnets.

Surveys like "[8] A Survey of Botnet and Botnet Detection Methods" provide a comprehensive overview, exploring various detection approaches, including honeynet-based solutions. "[9] A Survey on Botnets, Issues, Threats, Methods, Detection and Prevention" delves into the role of machine-based learning, specifically Auto-Encoders, in addressing botnet challenges. Studies like "[13] Machine learning DDoS detection for consumer internet of things" and "[14] A flow-based botnet detection using supervised machine learning" propose ML algorithms, including neural networks, for detecting DDoS attacks and distinguishing botnet traffic, respectively.

Network-based approaches, such as "[11] Network-based detection of IoT botnet attacks," leverage deep autoencoders to identify anomalous traffic from compromised IoT devices. Temporal evolution tracking, as explored in "[12] Tracking temporal evolution of network activity for botnet detection," addresses the challenge of evolving botnets, while "[15] Revealing botnet membership using DNSBL counter-intelligence" investigates DNS-based blackhole list (DNSBL) lookups for effective botnet membership identification. In conclusion, this literature survey, incorporating papers [2], [3], [4], [8], [9], [13], [14], [11], [12], and [15], highlights diverse ML approaches for botnet detection, paving the way for a deeper understanding of their strengths, weaknesses, and potential areas for further research.

Data Collection and Analysis

For successful detection of a botnet in real-time, it is necessary to first build a detection model in a test environment before deploying it for real-time applications. Most of the dataset available for botnet detection suffers from the problem like traffic obtained from simulated environment and creation of fake traffic that does not reflect real-time traffic. The main aim in botnet detection would be to have a real-time, not simulated one.

SDN Dataset

The SDN Dataset is a Botnet Traffic Dataset that was captured in September 2020. It is a Dataset which consists of Botnets and Normal Flow Packets.

The Simulation starts by creating ten topologies in mini-net in which switches are connected to single Ryu controller. Network simulation runs for benign TCP, UDP and ICMP traffic and malicious traffic which is the collection of TCP Syn attack, UDP Flood attack, ICMP attack.

A normal packet is a traffic that corresponds to traffic created by a naive user like opening mail inbox, surfing social media websites and scouring the internet for online resources.

A Botnet packet is a traffic that corresponds to traffic created by a malicious user like DDoS Attack, Spam E-Mails or malware spreading codes.

Table 1: Dataset Diversity Distribution

Total Flow	Normal Flow	Normal Flow (%)	Botnet Flow	Botnet Flow (%)
104300	63539	60.91	40761	39.08

Dataset Features

Total 23 features are available in the data set in which some are extracted from the switches and others are calculated.

Table 2: Feature Columns Description

Dt	Date and Time in Numerical Format
Switch	Switch number of the respective topology
Src	IPV4 Address of the Source
Dst	IPV4 Address of the Destination
Pktcount	No. of packets in the switch
Bytecount	No. of bytes in the switch
Dur	Simulation Time
Dur-nsec	Duration in (ns)
Tot-dur	Total Duration of Traffic in Network
Flows	Number of Flows
Packetins	Packet per Flow by Monitoring Interval
Pktperflow	Packet Count during a single Flow
Byteperflow	Byte Count during a single Flow
Pktrate	Number of packets sent per second
Pairflow	Packet count during a single flow
Protocol	For identifying TCP, UDP and ICMP traffic
Port-no	Port where packets were found
Tx-bytes	Number of bytes sent from the switch port
Rx-bytes	Number of bytes sent on the switch port
Tx-kbps	Data transfer rate
Rx-kbps	Data receiving rate
Tot-kbps	Sum of tx-kbps and rx-kbps
Label	Label (0) - Begign Traffic Label (1) - Botnet Traffic

Descriptive Analytics

The SDN dataset has total records of 104300 records out of which 60.91% traffic is normal traffic and 39.08% is botnet traffic. The distribution of traffic is shown in Figure 1 and it shows a significant balance that is present in the dataset. Protocol feature has 80.4% traffic using UDP protocol, followed by 18% of TCP protocol and 34% of ICMP protocol. The distribution of protocol is shown in Figure 2. The direction of the traffic was mainly bidirectional of 77.6 % followed by unidirectional with 21.8%.

Figure 1: Traffic Distribution

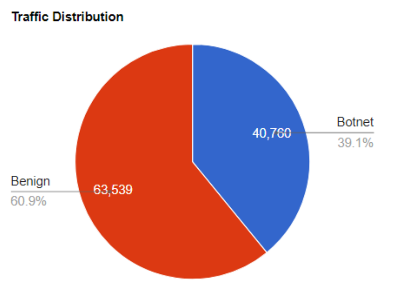
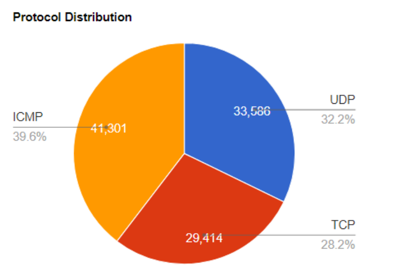


Figure 2: Protocol Distribution



Research Methodology

The field of machine learning is now the most widely implemented and experimented area. [10 – 15] have described various techniques of botnet detection using machine learning in combination with botnet characteristics. Botnet detection using machine learning techniques like Ramdom Forest (RF), Extra Trees (ET), Logistic Regression (LR), and Multinomial Naïve Bayes model based on DNS Query data is mentioned in [10]. Bots of the botnet receive code and commands from the C&C server by performing lookup queries generated using DGA or fast-flux. [10] identifies that the IP address of the C&C server is not a legit name and keeps on randomly changing to avoid detection. Also, the generated malicious domain names have characteristics like DNS, network, and lexical features entirely different from benign domain names. [10] approaches to solve the problem by collecting 16 vocabulary features from 2-g and 3-g clusters like mean, variance, standard deviation, entropy, consonants, vowels, number, and character characteristics and 2 characteristics from vowel distribution. IP addresses are random with datasets generated from Conficker, and DGA botnet (malicious) and top domain names from Alexa Internet (benign) (collection of domain names) in conjunction with machine learning models, [10] demonstrated the effectiveness of Random Forest machine learning model by delivering an accuracy of 98.71% in botnet detection.

Some related research papers to [10], [16] implemented techniques for detecting botnets on changing the IP addresses and it was successful in detecting traffic with many queries and terminated domains. Botnet detection models in the works of literature are heavily built for network and network devices as discussed in [10, 16, 17]. [14] considers Random Forest, Extra Trees, Logistic Regression and MNB and train these models on network flow parameters like length of the packet, size of the interval and protocol used. The detection pipeline is flow-based, uses either stateless or stateful features and is protocol-agnostic. [14] expresses the issues surrounding IoT devices like lightweight

Figure 3: Conceptual Map of Project

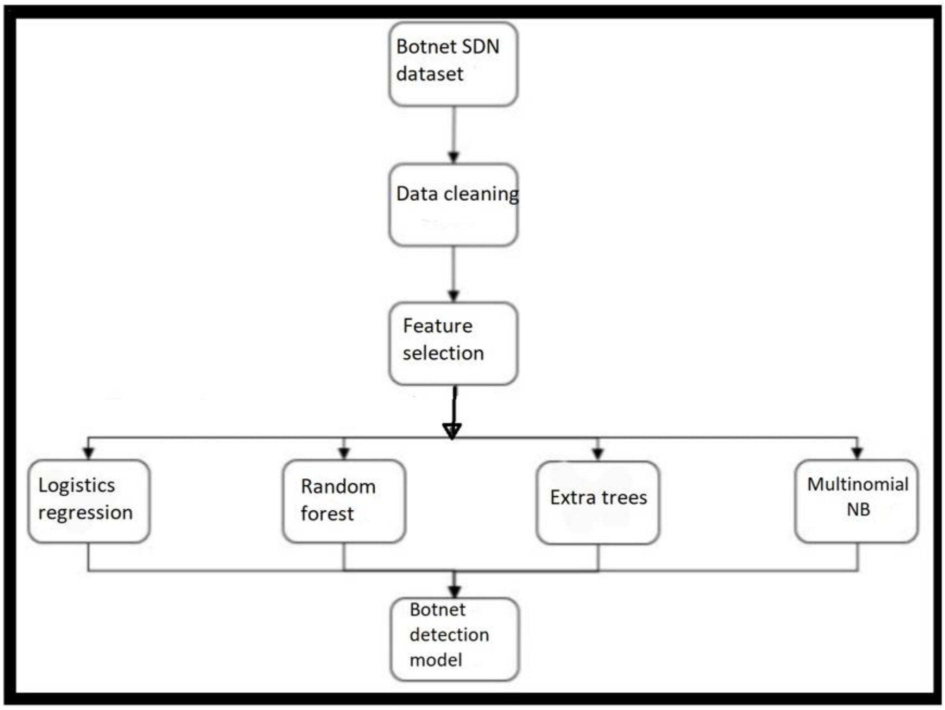
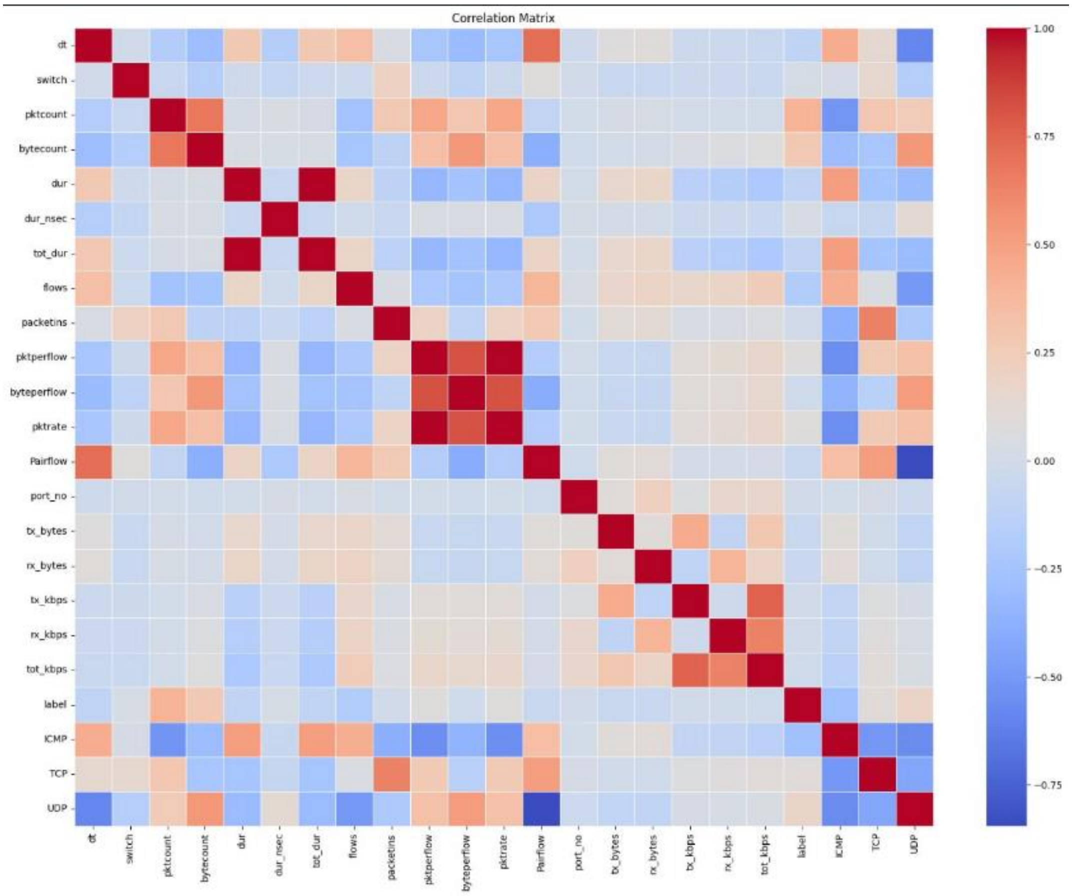


Figure 4: Correlation Matrix of Dataset Features



characteristics, limited memory, and computation power. Steps involved in building an IoT-based detection model is capturing the traffic, grouping the packets based on device and by time, extracting stateful and stateless features followed by a binary classification. [14] experimented with the dataset and found out that normal traffic packet size varies between 100 to 1200 bytes whereas attack traffic is under 100 bytes owing to repeated attacks. Also, the attack traffic has a lesser inter-packet interval in comparison to normal traffic. Most of the time protocols used during attack were TCP as opposed to UDP during normal situations. The classifiers were able to obtain training accuracy from 0.91 to 0.99. However, though the accuracy was 0.99, it was built on simulated data generated using DDoS network traffic-based botnet detection using machine learning. [14] predicts the model might be overfitting the data but is unsure of its performance on real-time attacks which opens the door for future research in the detection of botnets. Work was done in [11, 14] that focused on detecting botnet using IP address details and traffic flow characteristic, respectively.

On the other hand [16] focusses on leveraging the detection of botnet using an efficient flow-based technique by reducing the packet size and time of traffic flow under consideration. The model was developed to detect two P2P botnets namely Storm and Waledac botnets during the honeynet project. It has been noted in many papers that modern botnets are resilient to detection by employing techniques like protocols obfuscation, encrypted communication, fast-flux and random domain name generation using DGA. P2P botnets have a disastrous effect on industrial systems and their infrastructures. In order to train the models, [16] captured network traffic of five tuples like source IP address, source port, destination IP address, destination port and protocol used. As well for every traffic, 20 other statistical features were extracted. [16] employed batch analysis and limited analysis of the captured traffic by using eight different machine learning algorithms like Naïve Bayesian Classifier (NB), Logistic Regression (LR), Random Forest Classifier and Extra Trees Classifier. In the modelling process of botnet detection of [16], all MLA delivered impressive performance except Naïve Bayesian Classifier for both malicious as well as non-malicious traffic. The tree classifiers delivered promising classification performance, but Random Forest Classifier delivered the highest accuracy. Remaining MLAs experimented in [16], delivered poor performance for non-malicious traffic as compared to normal traffic since the dataset was skewed in the former case. Also, the initial 10 packets per flow are evident enough to detect botnet as opposed to monitoring the entire flow.

To summarize, [14] experimented with detecting botnet on consumer internet of thing device-based attack. [14] demonstrated the effectiveness of (MNB), Logistic Regression (LR), Extra Trees and Random Forest by achieving an average accuracy of 87%. [14] also considered stateful features like bandwidth, and IP destination address cardinality and novelty and stateless features like packet size, inter-packet interval and protocols separately and together during the training phase. [15] employed flow-based machine learning technique for botnet detection and experimented with models like Naïve Bayes, Bayesian Net, Artificial Neural Network, Support Vector Machine, Random Tree, Random Forest, and Decision Tree. The flow-based model was able to achieve accurate detection of traffic.

Experimental Results

1. Preprocessing Data

We have Dropped the unnecessary columns such as ‘Source’ and ‘Destination’.

Before Dropping Columns:

```
dt              0
switch          0
pktcount        0
bytecount       0
dur             0
dur_nsec        0
tot_dur         0
flows           0
packetins       0
pktperflow      0
byteperflow     0
pktrate         0
Pairflow        0
port_no         0
tx_bytes        0
rx_bytes        0
tx_kbps         0
rx_kbps       504
tot_kbps      504
label           0
ICMP            0
TCP             0
UDP             0
dtype: int64
```

After Dropping Columns:

```
dt              0
switch          0
pktcount        0
bytecount       0
dur             0
dur_nsec        0
tot_dur         0
flows           0
packetins       0
pktperflow      0
byteperflow     0
pktrate         0
Pairflow        0
port_no         0
tx_bytes        0
rx_bytes        0
tx_kbps         0
rx_kbps         0
tot_kbps        0
label           0
ICMP            0
TCP             0
UDP             0
dtype: int64
```

We have performed one-hot encoding on the column ‘Protocol’ to classify the categorical variable into a numerical format.

Before One-Hot Encoding:

Protocol
UDP
UDP
UDP
UDP
UDP

After One-Hot Encoding:

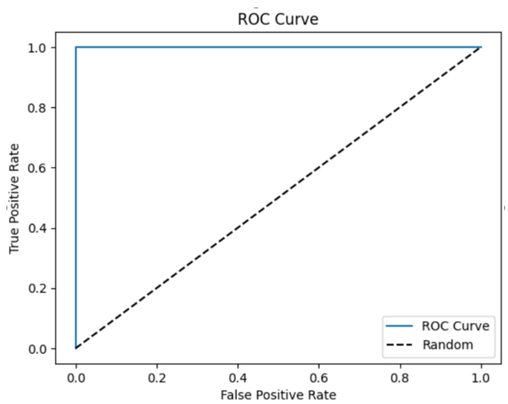
ICMP	TCP	UDP
1	0	0
1	0	0
1	0	0
0	1	0
1	0	0

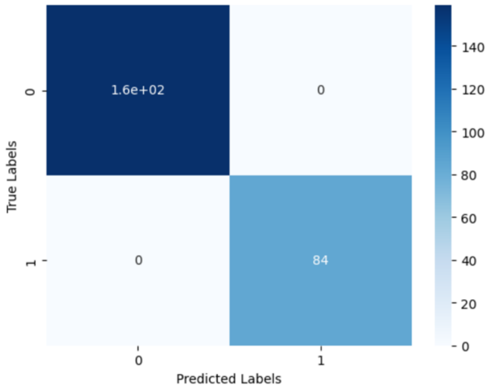
2. Random Forest

Random forest is an ensemble learning method that combines multiple decision trees to make predictions. Each decision tree in the random forest is built independently on a randomly sampled subset of the training data. The final prediction is obtained by aggregating the predictions of all the individual trees.

```
84 Bots found out of 243 packets
242
243
Accuracy is:  0.9958847736625515
```

```
AUC-ROC score: 1.0
```



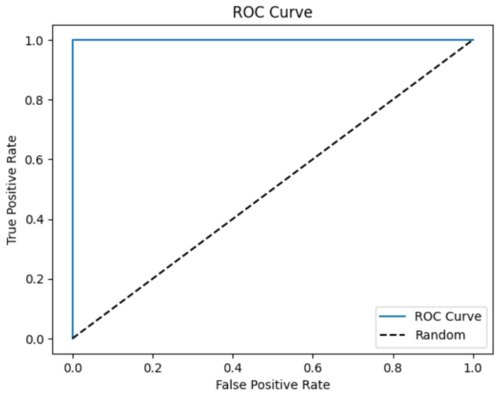


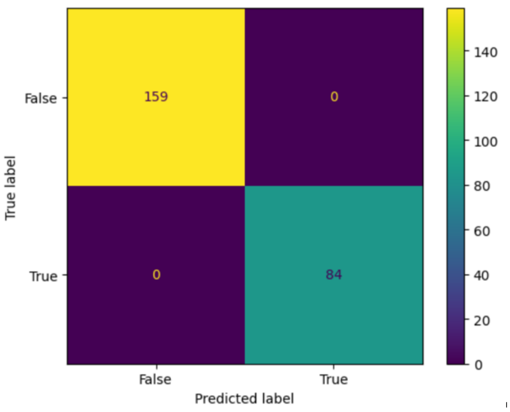
3. Extra Trees

Extra Trees is a variation of the random forest algorithm that introduces additional randomness during the construction of individual decision trees. In random forests, the algorithm considers a subset of features at each split point to find the best split.

```
84 Bots found out of 243 packets
242
243
Accuracy is:  0.9958847736625515
```

```
AUC-ROC score: 1.0
```



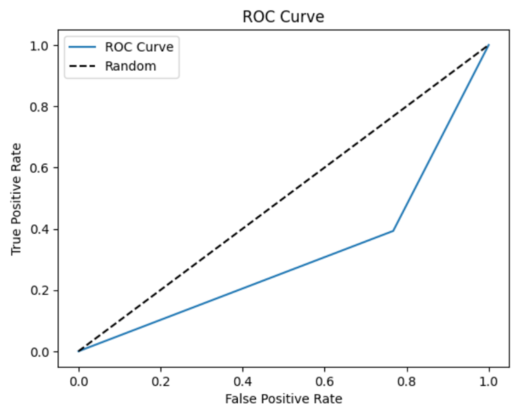


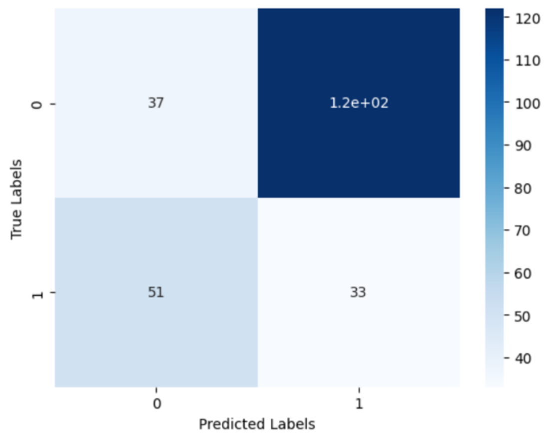
4. Logistic Regression

Logistic regression is widely used for binary classification, where the goal is to predict one of two possible classes based on input features. It can also be extended to handle multi-class classification problems through techniques like onevs-rest or softmax regression.

```
33 Bots found out of 243 packets
70
243
Accuracy is:  0.2880658436213992
```

```
AUC-ROC score: 0.312780772686433
```



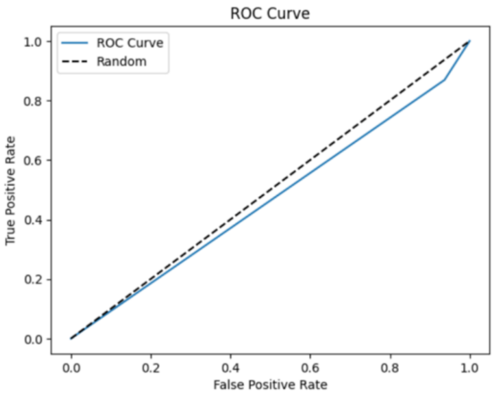


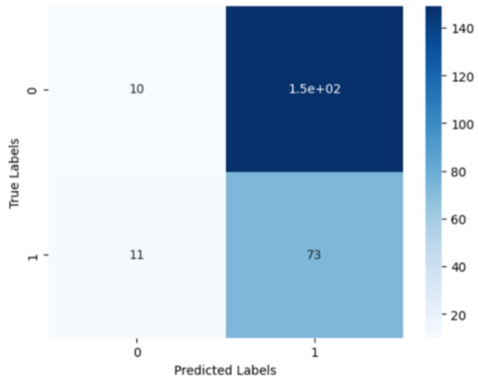
5. Multinomial Naïve Bayes

The multinomial naive Bayes algorithm is based on the principle of Bayes' theorem. It assumes that the features are conditionally independent given the class variable. It is called "naive" because it makes a strong assumption of feature independence, which may not hold true in many real-world scenarios. However, despite its simplifying assumptions, multinomial naive Bayes often performs well in practice, particularly in text classification tasks.

```
73 Bots found out of 243 packets
83
243
Accuracy is:  0.34156378600823045
```

```
AUC-ROC score: 0.46597035040431267
```





Results

Our data was used in building four models namely extra trees, random forests, logistics regression and multionomial NB. We calculated the accuracy, AUC-ROC score and the confusion matrix. The accuracy turned out to be 100%, 99.58%, 60.62% and and 56.62%. The data tested contained 243 rows of the same 22 features. The AUC-ROC value calculated is 1, 0.99, 0.28 and 0.33 respectively.

Conclusion

This research study has demonstrated that Extra Trees and Random Forest algorithms outperform Multinomial Naive Bayes (NB) and Logistic Regression models in the context of botnet detection. Through a comprehensive evaluation and comparison, it has been established that both Extra Trees and Random Forest exhibit superior in terms of accuracy (99.95 and 99.98 of extra trees and random forests percent when compared to 56.62 and 60.66 percent of logistics and multinomial), AUC-ROC score, and a better confusion matrix.

Extra Trees and Random Forest models are particularly adept at handling high-dimensional feature spaces, which is essential in botnet detection where numerous network-based attributes need to be considered simultaneously. These algorithms automatically select informative features and mitigate the impact of irrelevant or noisy attributes, thereby enhancing their ability to accurately distinguish between normal network traffic and botnet activity.

The reason for failure of logistic regression assumes a linear relationship between the predictor variables and the log-odds of the outcome. If the relationship is non-linear, logistic regression may fail to capture complex patterns accurately. Also, when the predictor variables are highly correlated, logistic regression can struggle to provide reliable coefficient estimates. This scenario can lead to issues such as multicollinearity and unstable model performance.

Multinomial Naive Bayes (NB) is designed to handle discrete features, such as word frequencies or categorical variables. When applied to numeric-heavy data, Multinomial NB may encounter challenges or fail to provide accurate results. Multinomial NB operates on the assumption of discrete features with discrete probability distributions. When dealing with numeric data, it requires binning or discretization of the numeric values. This binning process can result in a loss of information and granularity in the data. The discretization process may not capture the underlying distribution of the numeric features accurately, leading to suboptimal results.

Moreover, the efficiency of Extra Trees and Random Forest algorithms enables real-time or near real-time botnet detection, which is crucial for timely response and mitigation. The parallelization capabilities and inherent scalability of these algorithms make them well-suited for processing large volumes of network traffic data, thereby reducing detection latency and enhancing the overall effectiveness of botnet detection systems.

In conclusion, the results of this research highlight the superiority of Extra Trees and Random Forest algorithms over Multinomial NB and Logistic Regression for botnet detection. Their ability to handle complex patterns, high-dimensional data, imbalanced datasets, and their efficiency in real-time detection make them valuable tools for identifying and mitigating botnet activity in network environments. Future research can further explore the application of these algorithms in evolving botnet detection scenarios and investigate their potential for improving the accuracy and resilience of botnet detection systems.

References

1. Biau, Gérard & Scornet, Erwan. (2015). A Random Forest Guided Tour. TEST. 25. 10.1007/s11749-016-0481-7.

2. Biau, Gérard & Scornet, Erwan. (2015). A Random Forest Guided Tour. TEST. 25. 10.1007/s11749-016-0481-7.

3. L., Mary Gladence & Karthi, M. & Anu, Maria. (2015). A statistical comparison of logistic regression and different bayes classification methods for machine learning. ARPN Journal of Engineering and Applied Sciences. 10. 5947-5953.

4. Geurts, Pierre & Ernst, Damien & Wehenkel, Louis. (2006). Extremely Randomized Trees. Machine Learning. 63. 3-42. 10.1007/s10994-006-6226-1.

5. Xu, Shuo & Li, Yan & Zheng, Wang. (2017). Bayesian Multinomial Naïve Bayes Classifier to Text Classification. 347-352. 10.1007/978-981-10-5041-1_57.

6. Tyagi, Amit & Gnanasekaran, Aghila. (2011). A Wide Scale Survey on Botnet. International Journal of Computer Applications. 34.

7. Waiwnright, Polly & Kettani, Houssain. (2019). An Analysis of Botnet Models. 10.1145/3314545.3314562.

8. Xing, Ying & Shu, Hui & Zhao, Hao & Li, Dannong & Guo, Li. (2021). Survey on Botnet Detection Techniques: Classification, Methods, and Evaluation. Mathematical Problems in Engineering. 2021. 1-24. 10.1155/2021/6640499.

9. Owen, Harry & Zarrin, Javad & Shahrzad, Mahshid. (2022). A Survey on Botnets, Issues, Threats, Methods, Detection and Prevention. Journal of Cybersecurity and Privacy. 2. 74-88. 10.3390/jcp2010006.

10. Vishwakarma, Ruchi & Jain, Ankit. (2020). A survey of DDoS attacking techniques and defence mechanisms in the IoT network. Telecommunication Systems. 73. 10.1007/s11235-019-00599-z.

11. Meidan, Yair & Bohadana, Michael & Mathov, Yael & Mirsky, Yisroel & Shabtai, Asaf & Breitenbacher, Dominik & Elovici, Yuval. (2018). N-BaIoT—Network-Based Detection of IoT Botnet Attacks Using Deep Autoencoders. IEEE Pervasive Computing. 17. 12-22.10.1109/MPRV.2018.03367731.

12. Sinha, Kapil, Aruna Viswanathan and J. Bunn. "Tracking Temporal Evolution of Network Activity for Botnet Detection." *ArXiv* abs/1908.03443 (2019): n. pag.

13. R. Doshi, N. Aphtorpe and N. Feamster, "Machine learning DDoS detection for consumer internet of things devices", Apr. 11, 2018, [Online]. Available: https://arxiv.org/abs/1804.04159

14. A. Ramachandran, N. Feamster and D. Dagon, "Revealing botnet membership using

DNSBL counter-intelligence", in Proc. of the 2nd USENIX: Steps to Reducing Unwanted

Traffic on the Internet, San Jose, CA, USA, July 7, 2006, pp. 49–54

15. M. Stevanovic and J. Pedersen, "An efficient flow-based botnet detection using supervised machine learning", in 2014 Int. Conf. Comput., Netw. and Comm. (ICNC), 3-6 Feb. 2014,

DOI: 10.1109/ICCNC.2014.6785439.