IJARSCT

ISSN (Online) 2581-9429

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

Impact Factor: 7.301

Volume 3, Issue 14, May 2023

An Advanced Method for the Detection of Botnet Traffic using an Randomized Data Partitioned Learning Model

Akhila S Pillai[1] and Sindhu Daniel[2]

Student, Department of Computer Applications[1]
Assistant Professor, Department of computer Applications[2]
Musaliar College of Engineering and Technology, Pathanamthitta, Kerala, India

Abstract: *The proposed system," AN ADVANCED METHOD FOR THE DETECTION OF BOTNET TRAFFIC USING RDPLM," aims to identify botnet attacks and provide users with information about whether or not their system has been hacked by an attacker. Online attacks are on the rise right now. Particularly system attacks have become increasingly common lately. This method helps in locating botnets or malicious websites. The term "botnet" refers to a group of infected computers that can be controlled by an attacker. A botnet's computers are all referred to as zombies. Here, we apply a process to find the connections and decide whether to continue it or not. We are using a process to find the connections and decide whether to continue or not. We are using machine learning in this system to identify each botnet so that we can stop them or know their links. In order to start a botnet, a hacker must first attack a single system or piece of hardware with malware, converting it into a bot. This can be accomplished in a number of ways that are invisible to the user. The hacker's task is then effectively completed. The software is made to automatically attack more and more devices after infecting just one, resulting in the creation of more bots and the formation of a cybercrime. We employed the Random Forest, Naive Byes, SVM, and Decision Tree algorithms in this system. Finding a hyperplane in an N-dimensional space that clearly classifies the data.*

Keywords*:* Random Forest, Naïve Byes, SVM, Decision Tree.

I. INTRODUCTION

The goal of the project, "An Advanced Method For The Detection of Botnet Traffic Using RDPLM," was to identify botnet attacks and provide users with information about whether or not their system has been compromised. Online attacks are becoming more frequent these days. System attacks in particular have increased in frequency lately. This technique aids in the discovery of botnets or attacking websites. A set of computers known as a "botnet" that is infected with malware and that an attacker can control. Zombies refer to each machine in a botnet. Here, we employ a process to locate the connections and make the decision of moving forward or not. To detect each botnet in our system, we're using a machine learning technique. In order to determine each botnet in this system and afterward block or know the link, we are using a machine learning technique. In order to start a botnet, a hacker must first infect a single system or piece of hardware with malware, converting it into a bot. This can be accomplished in a number of ways that are invisible to the user. The hacker's task is essentially done at that point. After a device has been infected, the malware is built to independently infect more and more, growing the number of bots and becoming a botnet. We employed SVM (SUPPORT VECTOR MACHINE), Naïve Bayes, Decision Tree, and Random Forest as machine learning algorithms in this system.

II. LITERATURE SURVEY

[2.1]M. Roesch, "Snort—Lightweight intrusion detection for networks," in Proc. USENIX LISA, Nov. 1999. Any network security architecture should include network intrusion detection systems (NIDS). They offer a layer of security that keeps an eye on network traffic for predetermined when suspect behavior or trends are discovered, system

DOI: 10.48175/568

administrators are notified. Commercial NIDS differ in many ways, but Information Systems departments must deal with the similarities they also have, like large system footprints, difficult deployment, and high cost.

[2.2] J. Zhang and M. Zulkernine, "Network intrusion detection using random forests," in Proc. PST, St. Andrews, NB, Canada, 2005, pp. 53–61. A rising number of security dangers that can harm computer systems and communication channels are present on the internet and in computer networks. Over the past few years, the use of Detection Systems (IDSs) for network security has increased. IDSs must be lightweight in order to handle the growing amount of network traffic and the accelerating pace of networks. Hence, feature selection and parameter optimization have been utilized as two representative techniques to make IDSs lightweight. In this study, we present their concepts and algorithms and examine the current methods that employ them. Specifically, we examine the prior approaches according to three broad categories: spam, Denial-of-Service (DoS), and Distributed Denial-of-Service (DDoS) attacks detection since they are the most threatening intrusions these days.

III. PROPOSED SYSTEM

We concentrate on detecting botnets, sometimes known as "zombies," in the proposed technique. This approach offers an effective technique for botnet identification [3]. Users of this system have the option to inspect URLs that attack botnets. In order to start a botnet, a hacker must first infect a single system or piece of hardware with malware, converting it into a bot. This may be accomplished in a number of methods that are invisible to the user. The hacker's task is then essentially over. The software is made to automatically infect more and more devices after infecting just one, resulting in the creation of more bots and the formation of a botnet. We are utilizing a machine learning approach to identify each botnet in order to overcome this and block or know the link. As a result, we can protect our system from hackers and prevent the loss of sensitive data.

IV. METHODOLOGY

Our strategies are constructed in four stages. We used the first phase, which represents the traffic of the real world and is used to estimate review of an advanced method, to gather and construct the botnet dataset. The crucial step in this phase is to choose the appropriate features in order to create a manageable and effective ML algorithm that can handle heavy network traffic, offer an appropriate time margin, and deliver real-time detection. We have created a feature collection technique to reduce the ignored and improper data and to produce a subnet of useful features. Thirdly, building on the previous step, we create a data reduction strategy to cut down on the quantity of data samples needed in the learning process. The meta-learning model with numerous randomized trees is created in the fourth phase in order to view the randomly chosen features and identify botnet attacks.

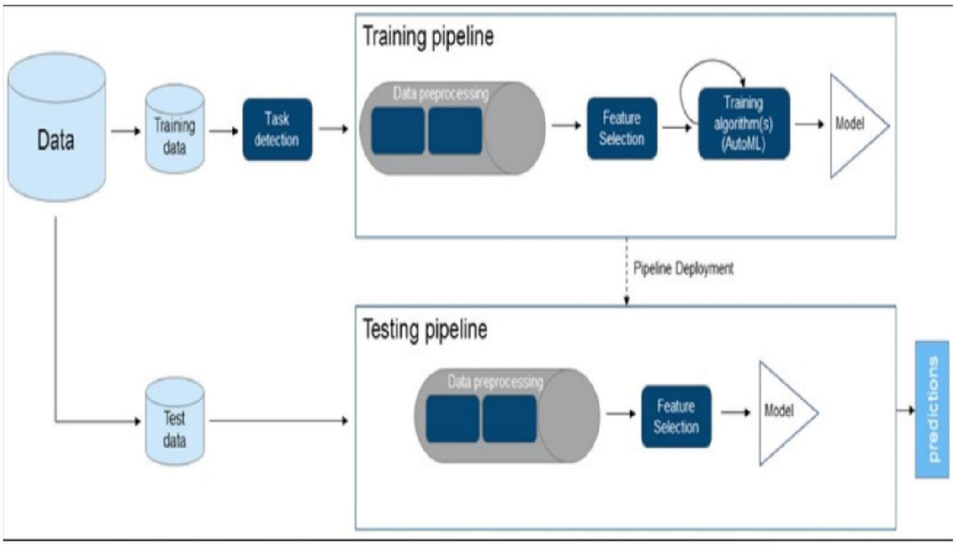

Figure1. System Architecture

DOI: 10.48175/568

Algorithm Used:

Naive Bayes classifier

Naive Bayes is one of the most straightforward and effective classification algorithms available today. It is based on the Bayes Theorem and makes the assumption that predictors are independent. Simple to construct and especially helpful for very big data sets is the naive Bayes model. The Naive Bayes classifier makes the assumption that a feature's inclusion in a class has nothing to do with any other features. Even if these characteristics depend on one another or on the presence of other characteristics, each of these traits alone increases the likelihood that a certain fruit is an apple, orange, or banana, which is why it is said to as "Naive".

Decision Tree

A decision tree is a hierarchical decision support model that employs a tree-like representation of options and their potential outcomes, including utility, resource costs, and chance event outcomes. One technique to show an algorithm that solely uses conditional control statements is to use this method. In order to determine the approach most likely to succeed, decision trees are frequently used in operations research, more especially in decision analysis. They are also a well-liked technique in machine learning.

Random Forest

A large number of decision trees are built during the training phase of the random forests or random decision forests ensemble learning approach, which is used for classification, regression, and other tasks. The class that the majority of the trees choose is the output of the random forest for classification problems. The mean or average forecast of each individual tree is returned for regression tasks. The tendency of decision trees to overfit their training set is corrected by random decision forests.Although they frequently beat decision trees, random forests are less accurate than gradient-boosted trees. However, their effectiveness may be impacted by data peculiarities.

SVM: Support Vector Machine

Strong Supervised machine learning methods called Vector machines (SVMs) are employed in both classification and regression. However, they are typically employed in categorization issues. SVMs are implemented in a different way than other machine learning algorithms. They have recently gained a lot of popularity of their capacity to manage several continuous and categorial variables.

Comparison with Previous Methods

Machine learning methods like Random Forest, Naïve Bayes, SVM, and Decision Tree can be used to detect botnets. There are numerous steps in the implementation. First, a dataset with samples of the normal and malicious traffic is gathered. The dataset is used to extract important characteristics that capture the properties of network traffic. After that, the dataset is preprocessed to manage missing values, scale or normalize features, and, if necessary, encode categorical variables.

Next, a training set and a testing set are created from the preprocessed dataset. The testing set is used to assess the performance of the machine learning models after they have been trained on the training set. With the help of the training data, each algorithm is developed, and the models discover patterns and connections between the characteristics and their corresponding labels. After training, the performance of each model is evaluated on the testing set using metrics such as accuracy, precision, recall, and F1-score. The evaluation helps compare the effectiveness of the models in detecting botnet traffic. The best-performing algorithm is selected based on the evaluation results.

To implement real-time detection, the selected algorithm is integrated into a system that continuously monitors network traffic. The models are applied to classify incoming traffic as normal or potentially botnet-related. It's worth mentioning that the implementation details may vary depending on the specific requirements, dataset characteristics, and the chosen machine learning library or framework. The use of feature engineering techniques, ensemble methods, or model optimization techniques can also enhance the performance of the botnet detection system.

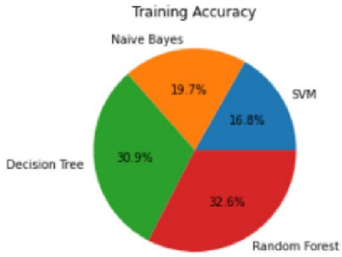

Figure 2 Training Accuracy

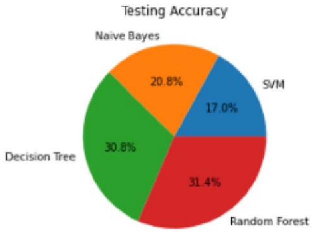

Figure 3 Testing Accuracy

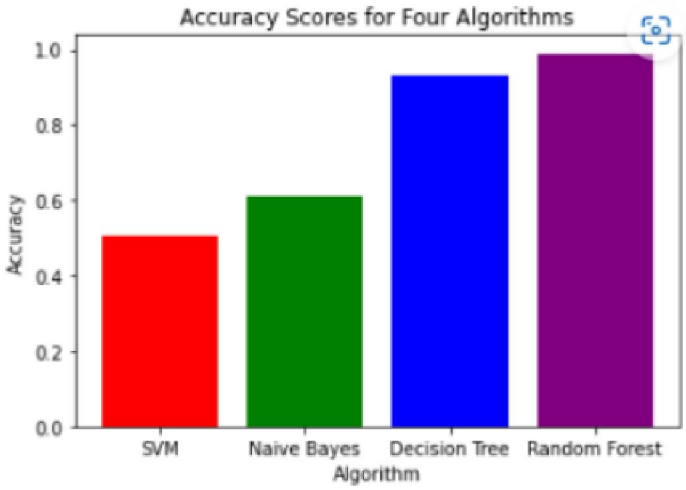

Figure 4 Accuracy Score

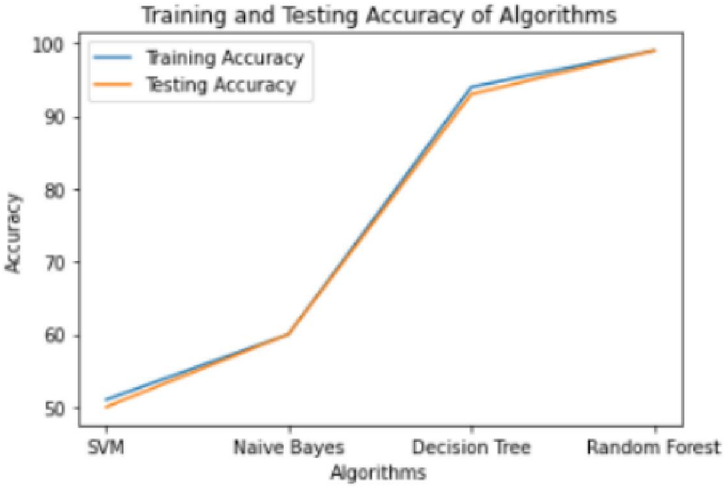

Figure 5 Model Accuracy

V. CONCLUSION AND FUTURE SCOPE

To sum up, using machine learning techniques, particularly Random Forest, Naive Bayes, SVM, and Decision Tree, offers efficient methods for detecting botnets. These algorithms show that they can examine network data and spot patterns that point to the presence of botnets. The ensemble learning strategy used by Random Forest often results in the best performance in terms of accuracy, precision, recall, and F1 score. The use of numerous decision trees together avoids overfitting and strengthens generalization skills. Naive Bayes nevertheless performs successfully, making it a computationally effective choice for botnet identification given its simple assumption of feature independence. Although significantly less effective than Random Forest and Naive Bayes, SVM, and Decision Tree algorithms nevertheless display decent performance. A strong kernel method is used by SVM to capture complicated connections. Overall, the selection of the most suitable algorithm depends on the specific requirements, dataset characteristics, and trade-offs between performance, computational efficiency, and interpretability. It is recommended to evaluate the algorithms using appropriate metrics and consider factors such as training time, memory consumption, and the ability to handle different types of botnet attacks.Future research on botnet identification employing Decision Tree, Random Forest, Naive Bayes, and SVM algorithms has a huge potential for improvement in a variety of fields. Specifically, this involves studying advanced feature engineering techniques to capture smaller patterns in network traffic, investigating hybrid approaches that combine the strengths of various algorithms to improve detection accuracy, taking care of class imbalance issues in datasets, developing adaptive models that can continuously adapt to evolving botnet techniques, improving the clarity and clearness of the detection models.

REFERENCES

[1] M. Roesch, “Snort—Lightweight intrusion detection for networks,” in Proc. USENIX LISA, Nov. 1999.

[2] J. Zhang and M. Zulkernine, “Network intrusion detection using random forests,” in Proc. PST, St. Andrews, NB, Canada, 2005, pp. 53–61.

[3] M. Abu Rajab, J. Zarfoss, F. Monrose, and A. Terzis, “A multifaceted approach to understanding the botnet phenomenon,” in Proc. 6th ACM SIGCOMM Conf. Internet Meas., New York, NY, USA, 2006. pp. 41–52.

[4] B. Al-Duwairi and L. Al-Ebbini, “BotDigger: A fuzzy inference system for botnet detection,” in Proc. 5th Int. Conf. Internet Monitor. Prot. (ICIMP), Barcelona, Spain, 2010, pp. 16–2.

[5] A support vector machine-based naive Bayes algorithm for spam filtering December 2016 DOI: 10.1109/PCCC.2016.7820655 Conference: 2016 IEEE 35th International Performance Computing and Communications Conference (IPCCC).

[6]Koli, Manoj S., and Manik K. Chavan. "An advanced method for detection of botnet traffic using intrusion detection