IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Survey on Botnet and its Detection Methods

Onkar kamble[1], Nikhil Tisangikar[2], Rutuja Galphade[3], Sakshi Gajare[4], Prof. Richa Agarwal[5]

[1]Information Technology Department, Savitribai Phule Pune University, KJ's Institute Trinity College of Engineering and Research, Pune, Maharashtra, India.
onkarkamble77@gmail.com[1]

[2]Information Technology Department, Savitribai Phule Pune University, KJ's Institute Trinity College of Engineering and Research, Pune, Maharashtra, India.
nikhiltisangikar2001@gmail.com[2]

[3]Information Technology Department, Savitribai Phule Pune University, KJ's Institute Trinity College of Engineering and Research, Pune, Maharashtra, India.
rutujagalphade1999@gmail.com[3]

[4]Information Technology Department, Savitribai Phule Pune University, KJ's Institute Trinity College of Engineering and Research, Pune, Maharashtra, India.
sakshigajare2002@gmail.com[4]

[5]Information Technology Department, Savitribai Phule Pune University, KJ's Institute Trinity College of Engineering and Research, Pune, Maharashtra, India.
richaagarwal.tcoer@kjei.edu.in[5]

Abstract - When multiple networks of bots emerged, the term "botnet" was coined. It is a collection of Internet-connected devices that run one or more bots. Botnets can be used to launch distributed denial-of-service attacks, send spam, and allow attackers to gain unauthorised access to network connections. The Owner (BotMaster) uses command and control software to manage the botnet. This paper discusses the accuracy of Botnet detection prediction using various models.

Keywords - Botnet, XGBoost, NaiveBayes, DDOS, Decision Tree, Random Forest, Network Traffic.

I. INTRODUCTION

A bot is an automated programme that runs on the internet; some run automatically, while others run when triggered by specific input. Internet-connected devices have been infected with bot software. These internet-connected devices are simply botnets. Following infection, these internet-connected devices follow the instructions given by the Botnet's owner, known as the Bot Master/Bot Herder, in four stages.

Following are the phases of the botnet infection:

Phase 1 Infection Initialization

A- Cybercriminals target "social media" posts, In the beginning, a cybercriminal will post a malicious link on social media websites, such as hoax advertisements, shammed icons, and so on. When users perform any action on these websites, their actions are proven to be incorrect, as the current page is redirected to a malicious website, where the software already planted by the BotMaster is installed.

B- Cybercriminals use the "infection method" approach. In this "Email Phishing" tactic, users are lured onto malicious websites by being redirected when a link is clicked, and their system is compromised.

C- "Email Attachments" cybercriminals disguise malicious software as an email, which is downloaded when clicked and infects the entire system.

Phase 2 Connection to C2C Server

The system establishes a connection with a command-and-control (C & C) server, which establishes unauthorised connections on a regular basis or may complete after infecting the system with malicious activity. Any infected machine communicating with a command and control server will agree to launch a coordinated attack. P2P, TELNET, and IRC are some examples.

Phase 3 Control

By installing botnets on compromised machines, a cybercriminal (BotMaster) supervises the command and control of botnets for remote process execution.

BotMasters use Tor/shells to conceal their identities by using proxies to disguise their IP addresses.

Phase 4 Multiplication

Botmasters use botnets to infect numerous internet devices in the first three phases, including fraud, spam emails, DDOS, keyloggers, and the Miria botnet. The most recent attack was "Kashmir Black," an active botnet that encompassed thousands of compromised systems in 30 countries and exploited dozens of vulnerabilities by targeting their CMS. It is believed that the "Kashmir Black" campaign began around the end of November 2019 and was designed to target CMS platforms such as Vbulletin, Opencart, Yeager, Joomla!, and WordPress. As a result of learning about these vicious internet attacks that occur on a daily basis. We decided to address this problem by implementing an ML model. In this paper, we will use our ML model to fill in the gaps

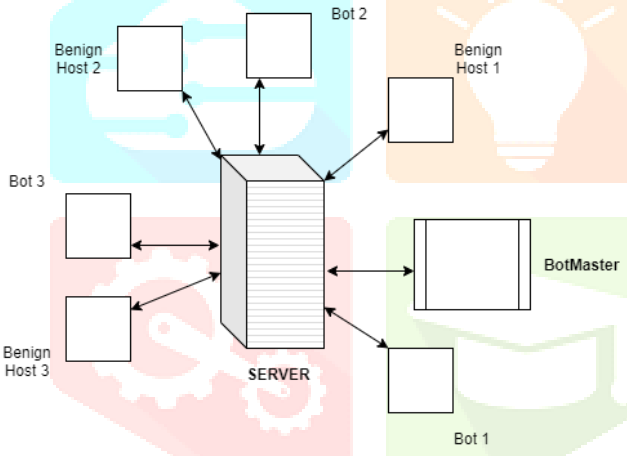

Fig 1.Model of Botnet

II. METHODS AND APPROACHES

In order to detect a botnet we must apply the correct method and follow feasible approaches.

A. XGBoost

Boosting is a sequential technique which follows the principle stated by the ensemble model. It has a set of weak learners which helps to ameliorate prediction accuracy. at any instant s model outcomes are weighed on previous instant t-1. outcomes which get predicted correctly assigned as a lower weight and which got miss-classified weighted higher.

Four classifiers (shown above in four boxes) are attempting to classify bots B and Benign Host H as uniformly as possible.

and vulnerabilities on the canvas. To understand the vastness of Machine Learning models, consider the basic Botnet model. Figure 1 depicts a basic botnet model in which the botmaster is directly or indirectly connected to all other entities such as servers, bots, and benign hosts via two-way communication.

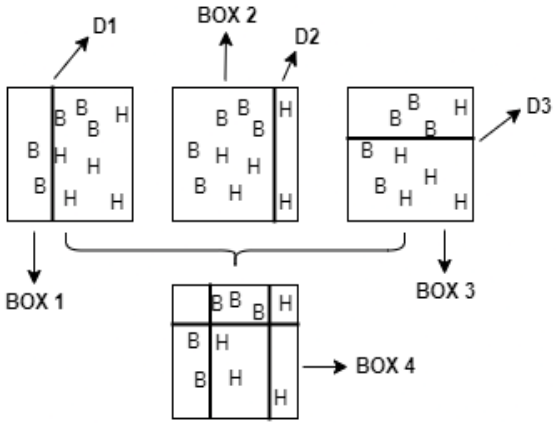

Fig 2.XGBoost

1. Box 1: The first classifier (a decision stump) draws a vertical line (splits it) D1. Everything to the left of D1 is B, and everything to the right of D1 is H. This classifier, however, misclassified three B points. Decision Stump is a Decision Tree model that only divides at one level, so the final prediction is based on a single feature.

2. Box 2: The second classifier emphasises the three B misclassified points (note the larger size of B) and draws a vertical line at D2. Again, anything to the right of D2 is H, and anything to the left is B. Nonetheless, it makes errors by incorrectly classifying three H points.

3.Box 3: Once again, the third classifier emphasises the three H misclassified points and draws a horizontal line at D3. Nonetheless, the classifier fails to correctly classify points.

4Box 4: Combination of weak weighted classifiers (Box 1, Box 2 and Box 3). It does a good job of correctly classifying all points. This is the basic idea behind how this algorithm will assist us in identifying botnets.

B. Naive Bayes Algorithm

The Naive Bayes classifier is a probabilistic machine learning model for classification. The classifier's core is based on the Bayes theorem P(A|B)=(P(B|A)P(A)/P. (B)

The Naive Bayes classifier is a type of probabilistic machine learning model used for classification tasks. The Bayes theorem is at the heart of the classifier.

It is primarily used in sentiment analysis, spam filtering, and other applications. Naive Bayes is quick and simple to implement, but it has the

drawback of requiring the predictors to be independent. In most real-world cases, the predictors are dependent, which reduces the classifier's performance.

C. Decision Tree Algorithm

In the field of machine learning, Learning and prediction are two steps in the classification process. In learning steps, the model is built on given training data. In the prediction step, the model is used to predict the response for given data. The Decision Tree Algorithm (DTA) is a type of Supervised Learning Algorithm (SLA). Unlike other SLA, DTA can be used to solve regression and classification problems. The goal of using a Decision Tree is to learn simple decision rules inferred from prior data to create a training model that can be used to predict the class or value of the target variable (training data). In Decision Trees, we begin at the root and work our way up to the prediction of a class label. We compare the values of the root attribute and the attribute of the record. Based on the comparison, we follow the branch corresponding to that value and proceed to the next node.

D. Random Forest

Random Forest is a well-known machine learning algorithm that belongs to the supervised learning technique. In Machine Learning, the Random Forest Algorithm can be used for both classification and regression problems. Random forest is based on ENSEMBLE LEARNING, which is a process that involves combining multiple classifiers to solve a complex problem and improve the model's performance. In layman's terms, Random Forest employs Decision trees in a randomised fashion. When compared to other algorithms, Random Forest requires less training time and produces high accuracy output. Random forest performs well even with large datasets. Implementation Steps for random forest are as 1.Data Pre-processing

2.Fitting R.F algorithm to the Training set.

3.Predicting the test result.

4.Test accuracy of the result.

5.Visualizing the test set result.

ML Approach

Machine learning has a wide range of applications and methods for dealing with real-world problems in discrete domains. This is possible because of the abundance of data spread across the network, significant advancements in ML techniques, and advancements in computing capabilities. The components used to build a robust ML model for a given networking model are depicted in the figure. ML has been used to solve real-world complex problems in network operations and other fields due to its adaptability.

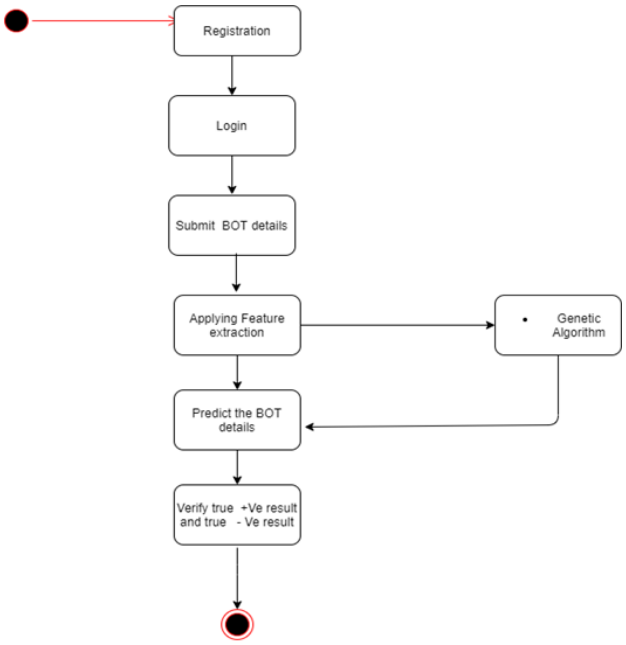

Fig 3. ML based solution.

In our survey, we discovered that perplexed problems across various network technologies can be solved by utilising various ML techniques, which is in accordance with the diverse applications of Machine Learning. In our paper, we discussed networking fragments such as QOE, QOS management, traffic prediction, congestion control, routing, and classification management to gain insights, scientific challenges, and the extent of ML in networking. Every effort is accountable and bears responsibility for breaking down the barriers to automatic network operations and their activities by utilising ML features in networking.

III. RESULTS AND DISCUSSION

We present a machine learning-based botnet detection system that has been demonstrated to be effective in detecting P2P botnets. Using a convolutional version of effective flow-based features, we extract convolutional versions and train a classification model. Artificial neural network with feed-forward. The experimental results show that detection accuracy using convolutional features is higher than detection accuracy using traditional features. On known P2P botnet datasets, it achieves 94.7 percent detection accuracy and 2.2 percent false positive rate. Furthermore, our system provides additional confidence testing to improve botnet detection performance. It also categorises network traffic based on insufficient confidence in the neural network. The experiment shows that this stage can boost detection accuracy to 98.6 percent while lowering false positive rates to 0.5 percent.

STEP 1

At this step the available options will help us to choose the better and feasible model to detect botnets.

Fig 6.Algorithms

STEP 2

First, we select the Algorithms that will be used to train our model using the KDD Cup Dataset during the training phase. After selecting the algorithm and training the model, we proceed to the Botnet detection page's drop down list. We choose the algorithm from the drop-down menu. To detect a Botnet, the system must be given a dataset with a large number of fields containing information from network logs and traffic.

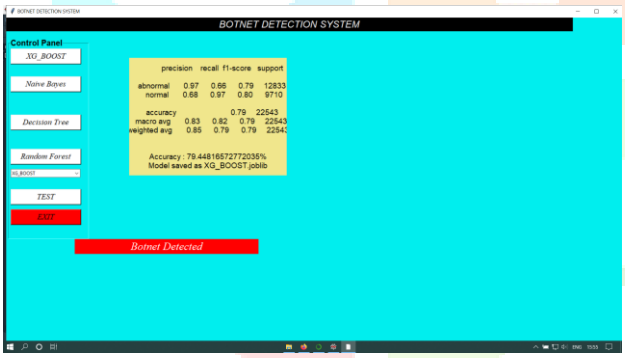

Fig 7. Result

Comparative analysis of botnets with different ML techniques gives us the idea that using a single The model for detecting botnets is no longer useful as technology advances and bots become smarter. So relying on a single model is a bad idea. We implemented various models in order to demonstrate the different prediction accuracy achieved by using various algorithms.

XGBoost- 79.448%

Naive Bayes- 45.029%

Decision Tree-79.532%

Random Forest-76.227%

IV. FUTURE SCOPE

We intend to scale up this model by improving its detection to work in real time. The proposed model detects the botnet but cannot handle large datasets. Real-time interpretation of n/w logs and monitoring of accuracy is still a work in progress. As technology advances, more and more new tools for handling large amounts of data with high accuracy in predicting botnet detection will become available.

V. CONCLUSION

This paper compares different algorithms and their accuracies to examine various techniques and methods for dealing with botnets in various situations across different networks. The main threat in bot detection is avoiding any loopholes or vulnerabilities in our own system while tracking them in order to terminate the bot's network before their botmaster achieves their vicious goal. We successfully implemented various models and achieved higher accuracy in predicting the presence of a Botnet in a system. The model is trained with 80 percent of the data and tested with the remaining 20 percent.

VI. REFERENCES

[1] Sudipta Chowdhury1*, Mojtaba Khanzadeh1, Ravi Akula1, Fangyan Zhang2, Song Zhang2, Hugh Medal1, Mohammad Marufuzzaman1, Linkan Bian1" Botnet detection using graph-based feature clustering".

[2] Zhuang and J. M. Chang, "PeerHunter: Detecting peer-to-peer botnets through community behavior analysis,".

[3]S. Lagraa, J. François, A. Lahmadi, M. Miner, C. Hammerschmidt and R. State, "BotGM: Unsupervised graph mining to detect botnets in traffic flows," 2017 1st Cyber Security in Networking Conference (CSNet), Rio de Janeiro, 2017, pp. 1-8, doi: 10.1109/CSNET.2017.8241990.

[4] Sara Khanchi, Ali Vahdat, Malcolm I. Heywood, A. Nur Zincir-Heywood,"On botnet detection with genetic programming under streaming data label budgets and class imbalance",Swarm and Evolutionary Computation,Volume 39,2018,,ISSN 2210-6502

[5] Jeeyung Kim, Alex Sim, Jinoh Kim, Kesheng Wu," Botnet Detection Using Recurrent Variational Autoencoder".

[6] Hagan, M., Kang, B., McLaughlin, K., & Sezer, S, "Peer Based Tracking using Multi-Tuple Indexing for Network Traffic".

[7] Raouf Boutaba 1, Mohammad A. Salahuddin 1, Noura Limam 1, Sara Ayoubi 1, Nashid Shahriar 1, Felipe Estrada-Solano1,2 and Oscar M. Caicedo 2 "Survey on machine learning for networking: evolution, applications and research opportunities".

[8]E. M. Hutchins, M. J. Cloppert, and R. M. Amin, "Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains," Inf. Warfare Security Res., vol. 1, no. 1, p. 80,2011.

[9]S. Chen, Y. Chen and W. Tzeng, "Effective Botnet Detection Through Neural Networks on Convolutional

Features," 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), New York, NY, 2018, pp. 372-378, doi: 10.1109/TrustCom/BigDataSE.2018.00062.
[10]B. Alothman and P. Rattadilok, "Towards using transfer learning for Botnet Detection," 2017 12th International Conference for Internet Technology and Secured Transactions (ICITST), Cambridge, 2017, pp. 281-282, doi: 10.23919/ICITST.2017.8356400.
[11]G. Vormayr, T. Zseby and J. Fabini, "Botnet Communication Patterns," in IEEE Communications Surveys & Tutorials, vol. 19, no. 4, pp. 2768-2796, Fourthquarter 2017, doi: 10.1109/COMST.2017.2749442.
[12]H. Dhayal and J. Kumar, "Botnet and P2P Botnet Detection Strategies: A Review," 2018 International Conference on Communication and Signal Processing (ICCSP), Chennai, 2018, pp. 1077-1082, doi: 10.1109/ICCSP.2018.8524529.
[13]C. Czosseck, G. Klein and F. Leder, "On the arms race around botnets - Setting up and taking down botnets," 2011 3rd International Conference on Cyber Conflict, Tallinn, 2011, pp. 1-14.
[14]K. Alieyan, M. Anbar, A. Almomani, R. Abdullah and M. Alauthman, "Botnets Detecting Attack Based on DNS Features," 2018 International Arab Conference on Information Technology (ACIT), Werdanye, Lebanon, 2018, pp. 1-4, doi: 10.1109/ACIT.2018.8672582.
[15]W. Zhang, Y. -J. Wang and X. -L. Wang, "A Survey of Defense against P2P Botnets," 2014 IEEE 12th International Conference on Dependable, Autonomic and Secure Computing, Dalian, 2014, pp. 97-102, doi: 10.1109/DASC.2014.26.
[16]W. Sun and H. Gou, "The Botnet Defense and Control," 2011 International Conference of Information Technology, Computer Engineering and Management Sciences, Nanjing, Jiangsu, 2011, pp. 339-342, doi: 10.1109/ICM.2011.218.

VII. REFERENCES

https://pure.qub.ac.uk/en/publications/peer-based-tracking-using-multi-tuple-indexing-for-network-traffi

https://arxiv.org/abs/2004.00234v1

https://jisajournal.springeropen.com/articles/10.1186/s13174-018-0087-2

https://scholar.nycu.edu.tw/en/publications/effective-botnet-detection-through-neural-networks-on-convolution

https://ieeexplore.ieee.org/document/8356400

https://www.researchgate.net/publication/331197841_Botnet_Detection_Techniques_and_Research_Challenges