STM JOURNALS

Research & Reviews: Journal of
Embedded System & Applications

ISSN: 2395-6712 (Online)
ISSN: 2321-8533 (Print)
Volume 12, Issue 2, 2024
May–August
DOI (Journal): 10.37591/RRJoESA

https://journals.stmjournals.com/rrjoesa

Review RRJoESA

Botnet Beacon: Unveiling Covert Networks with Advanced AI Detection Strategies

Kapil Kumar[1,*], Manju Khari[2]

Abstract

Securing information technology systems is paramount in today's interconnected world, where the reliability and security of networks and applications are of utmost importance. In this context, the development of a Botnet Detection System (BDS) that harnesses the power of AI classification algorithms becomes a critical endeavor. The primary objective of this work is to construct a comprehensive framework for a BDS that can efficiently gather network data and subject it to rigorous analysis using AI algorithms. To achieve this objective, the authors have chosen to utilize the botnet dataset, a widely recognized benchmark in the field, for training the classifier. This dataset serves as a rich source of information containing various network traffic data, particularly focusing on essential features that are crucial for attack classification. By employing this dataset, the authors ensure that the BDS is trained on a diverse set of network behaviors and attack patterns, enabling it to recognize and differentiate between legitimate and malicious activities effectively. The performance of the BDS is evaluated through a rigorous assessment, encompassing metrics such as accuracy, precision, and detection rate. These metrics are essential in gauging the BDS's ability to correctly identify and classify botnet activities while minimizing false positives. By conducting this comprehensive evaluation, the authors aim to ensure that the BDS is not only capable of detecting botnets but does so with a high degree of accuracy and reliability.

Keywords: Botnet detection system (BDS), artificial neural network (ANN), software-defined networking (SDN), machine learning, information security

INTRODUCTION

A dynamic programmer or someone attempting to access system resources that are prohibited from use is considered an intruder in a botnet detection system. A botnet is an attempt to breach the three primary pillars of security: confidentiality, integrity, and availability of computer resources, according to Alhakami [1]. A Botnet Detection System is a system that is designed to continuously monitor and identify any activity that compromises confidentiality, integrity, or availability. When a botnet detection system detects unauthorized access to a computer system across a network, it can take appropriate action in accordance with security standards.

***Author for Correspondence**
Kapil Kumar
E-mail: kapil.sharma0942211@gmail.com

[1]Research Scholar, School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi, India
[2]Professor, School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi, India

Received Date: July 17, 2024
Accepted Date: July 23, 2024
Published Date: July 26, 2024

Citation: Kapil Kumar, Manju Khari. Botnet Beacon: Unveiling Covert Networks with Advanced AI Detection Strategies. Research & Reviews: Journal of Embedded System & Applications. 2024; 12(2): 26–32p.

According to Almseidin *et al.*, computer hosts and botnet detection systems are managed by security-operating centres as attack countermeasures. Commercial botnet detection solutions are widely accessible these days [2]. This commercial implementation is generally weak and ineffective, which highlights the need for additional study on the dynamic botnet detection system. As a result, we will conduct research in the context of botnet detection systems that rely on machine

learning. The botnet is determined to be the most hazardous threat, posing the greatest risk of information loss to private, public, and government entities.

It is discovered that an intrusive activity is a dynamic action that tries to gain access to system resources that are prohibited from being utilised. A botnet is an attempt to breach the confidentiality, integrity, and availability, the three primary pillars of information security. The attacks are identified by using the botnet detection system (BDS). Consequently, we must create a model to reduce the risk of assaults and identify them using botnet dataset and AI classification algorithms. The project aims to create a better BDS for attack detection. One of the goals is to use the best AI classification algorithm to develop a BDS framework for attack detection. Attack detection accuracy has to be improved. to identify fresh or unfamiliar assaults.

PROBLEM STATEMENT

The work's primary technological problems can be described as follows.

1. The main issue is that attacks are incorrectly categorized due to their poor accuracy and precision. The main causes of this are an unequal distribution of attack categories and a wide variety of attack types that have similar traits.
2. Furthermore, it becomes apparent that the location of the BDS system, responsible for collecting data from the network payload, is at the crux of the problem. Any alterations to the gathered data can lead to inconsistent results, as malicious actors possess the capability to manipulate this data during its transit to the BDS.
3. Finally, the third problem relates to insufficient data collection techniques that depart from an exhaustive survey cycle.

RESEARCH OBJECTIVES

The work's primary technological contributions can be summed up as follows:

1. Improving Attack Classification Accuracy.
2. To balance the distribution of attack categories in the dataset, use techniques like under sampling and oversampling.
3. Examine cutting-edge machine learning algorithms that have improved ability to discriminate between similar attack types.
4. Update the classification model frequently with fresh information, then retrain it to take into account changing attack patterns.

This is the format for the remainder of the paper. Next Part reviews the literature and analyzes the limitations and current research. A diagram with multiple phases is used to describe the proposed work in the Section following that. The process of creating proposed models for the system and analyzing the results are explained in the Section after that. A summary is given in the last Section.

LITERATURE REVIEW

In their paper, Iraqi and El Bakkali are attempting to reduce the high non-response rate and false alerts by developing a botnet detection system utilizing a data mining approach [3]. According to the authors, the machine-learning algorithm used in the network anomaly; detection system is more efficient and effective than other ML techniques. This algorithm solves the Centralization problem. Lai *et al.* hope to disseminate Botnet detection classification for the proposed wireless network and control system [4]. The goal of the research is to use recurrent neural networks for binary and multiclass classification when all features are used, resulting in botnet detection systems with high accuracy. The research focuses on the development of a botnet detection system by the application of deep learning architecture and long short-term memory to recurrent neural networks, together with BDS training using KDD cup_99. Min *et al.* presented a study in which they constructed anomaly detection and classification using the supervised algorithm for learning [5]. By mixing supervised and unsupervised

learning, the authors hope to increase the system's accuracy and performance. Yan and Han presented the study, which combines application-level and unsupervised outlier detection techniques to detect botnet activity in Java applications for the enterprise [6]. In their paper, Woodiss-Field *et al*. describe experiments examining the efficacy of several traditional botnet detection techniques, such as BotMiner, BotProbe, and BotHunter, when used in conjunction with Internet of Things-based botnets [7]. They evaluated these methods' performance on traditional and IoT-based networks by running tests in a variety of simulated environments with internal network traffic generation tools. Variations in the total number of hosts, the frequency of abnormal activities, the types of botnet command and control (CnC), and the prevalence of infected hosts were noted during the simulations. Additionally, each botnet detection method's performance was assessed and validated using datasets sourced from outside sources. The 2024 study by Lagraa *et al.* provides a thorough examination of graph models that are used to represent, store, and visualize network security data [8]. It also performs an analysis of the techniques and algorithms used in data analysis. It also lists important graph features for botnet monitoring and detection in network security analytics. Additionally, the paper explores the difficulties and limitations that come with using graph-based techniques in network security and suggests possible directions for further research. This survey paper is essentially a useful resource for network security researchers and practitioners who want to use graph-based techniques to examine and detect malicious activity in networks. Wu *et al.* examine P2P botnet structural features via the prism of complex graph theory in their paper [9]. They present PeerG, an integrated representation learning and graph contrastive learning detection technique for P2P botnets. In a similar vein, Wei *et al.* suggest a brand-new, adaptable, and lightweight Network Intrusion Detection System (NIDS) intended to detect botnet activity in Internet of Things networks [10]. Their method uses a two-stage framework that is limited to features that are accessible at packet length, which allows for deployment on devices with limited resources and effective real-time operation. Moorthy and Nathiya focus on botnet malware detection by examining malicious packet net flows [11]. They use datasets of botnet attacks from repositories such as Information Security and Object Technology (ISOT) and Czech-University (CTU-13). The research computes network packets and makes use of bidirectional net flow data. To train and test the dataset, several algorithms are used, such as multi-layer perceptron, decision trees, and Support Vector Machines (SVM). After training and testing, the decision tree model achieves 92%, showing strong performance metrics and accuracy. This model is considered the best fit and helps identify malicious packets. Creating an alert system that alerts users when a malware packet enters a network is the main goal of the research. The authors of this study compare and analyze important recent developments in botnet detection, as described by Xing *et al*. [12]. They provide a classification of detection techniques in addition to delving into the architectural characteristics, life cycle, and command and control channels of botnets. The study highlights the application of state-of-the-art technologies to improve botnet detection capabilities, including deep learning, software-defined networking (SDN), swarm intelligence, complex networks, and moving target defense (MTD).

The primary goal of their paper, according to Shinan *et al.*, is to evaluate botnet detection methods using the principles of systematic reviews and meta-analyses (PRISMA) [13]. The reviewed a number of papers from 2015 in the area of SDN and from 2006 in the area of machine learning-based botnet detection. Authors specifically used highly regarded journals with the highest impact factors. The goal in this paper is to provide more detail on a number of research topics related to SDN, machine learning, detection methods, and botnet attacks. The difficulties facing current research and suggest avenues for further investigation. In their paper, Shi and Sun present a technique for identifying potential botnets by analyzing the behavioral patterns within network traffic derived from network packets [14]. Initially, packets are sampled over intervals, and behavioral characteristics are extracted from consecutive packets. Through the analysis of these features using novel deep learning models, the presence of botnet threats can be detected and categorized. Additionally, Ibrahim *et al.* propose a multi-layered framework for botnet detection employing machine learning algorithms [15]. This framework includes a filtering module and a classification module designed to pinpoint the command and control server of the botnet.

The authors outlined several requirements for our framework, among them are that it must be independent of protocol, independent of structure, and able to identify botnets even when encapsulated techniques are employed.

PROPOSED WORK

An artificial neural network algorithm based on deep learning principles is used in the suggested system. This algorithm consists of a series of steps with multiple layers that communicate information through three different layers to connect data: the input layer receives the data and preprocesses it by establishing statistical relationships between its variables and features to produce accurate results. The steps involved in using an Artificial Neural Network (ANN) for Botnet Detection Systems (BDS) are outlined in a structured framework. Python is the programming language used for dataset preprocessing, and the KDD_Cup99 dataset is used for analysis. Preprocessing is done on the dataset to remove duplicate entries and separate the independent and dependent variables. The classifier is then trained using training data, and feature extraction techniques are used to expedite the training process. The performance of the model and prediction accuracy are then assessed using test data. The suggested BDS framework with an ANN to identify attacks is depicted in Figure 1, where the classifier is trained and tested using data from a botnet.

DATA DESCRIPTION

Downloading the dataset from Kaggle is possible by using this link: https://www.kaggle.com/siddharthm1698/ddos-botnetattack-on-iot-devices. This dataset, which has many features that are useful for predicting malicious packets and performing in-depth analyses, provides a comprehensive collection of DDoS Botnet attacks executed by IoT devices. The dataset itself is highly skewed, which makes it difficult to balance its distribution even though it offers an alluring opportunity for exploration. However, this dataset is useful for gaining insights into malware behavior for security enthusiasts. It includes real-world examples of IoT devices being compromised and used in botnets to launch DDoS attacks, offering important new perspectives on the dynamic threat environment. There is extensive data on targets, traffic patterns, attack vectors, and durations. Using this dataset, researchers and security experts can create strategies and countermeasures that are effective in protecting IoT ecosystems.

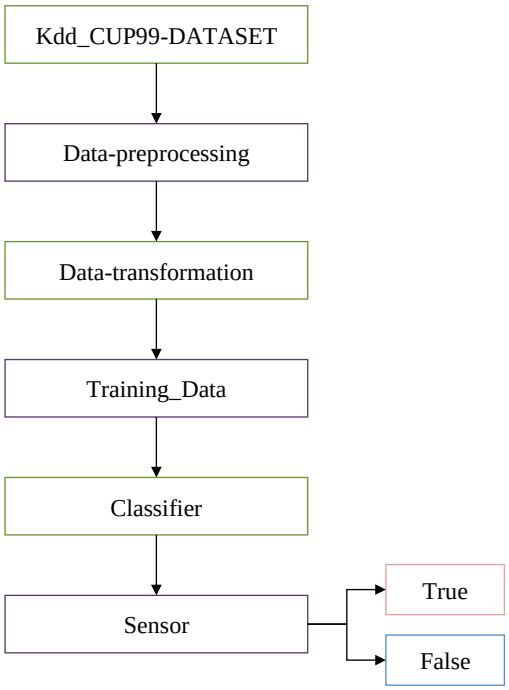

Figure 1. Projected framework.

It is advised that those who want to improve their IoT security knowledge enroll in an IoT certification program provided by a respectable organization like The IoT Academy. Attending a course like this makes participants valuable assets in the field of IoT cybersecurity by arming them with the knowledge and abilities needed to protect IoT devices and networks against new threats like DDoS botnet attacks.

RESULT AND ANALYSIS

The classification report for each model, which is produced from the SVM, Random Forest, and Naïve Bayes machine learning algorithms' performance evaluations, presents the best classifier for classification. Regression and classification tasks are two applications for the effective supervised learning algorithm Support Vector Machine (SVM). To efficiently divide various classes, SVM works by identifying the best hyperplane inside the feature space [16]. It seeks to maximize the margin between the closest data points, or support vectors, and the hyperplane in order to improve its capacity to generalize to new data. Using a variety of kernel functions, including polynomial, linear, and radial basis function (RBF), SVM can handle both linear and non-linear decision boundaries. ANN, a computational model derived from biological neural networks, represents an additional classification algorithm [17].

ANNs are made up of linked nodes arranged into three layers: output, hidden, and input. Each neuron receives input signals, weights them, performs a non-linear transformation known as an activation function, and then sends the result to neurons in the layer below. ANNs are capable of processing non-linear decision boundaries by comprehending intricate patterns and correlations within data. Neural networks trained with multiple hidden layers are able to learn hierarchical representations of data, a capability of ANNs known as deep learning. The Naive Bayes method of probabilistic classification relies on the assumption of feature independence and the Bayes theorem. It utilizes input features to calculate the probability of each class and determines which class has the highest probability to be the final prediction. Naive Bayes is a well-liked option for large-scale and real-time applications due to its well-known simplicity and effectiveness [18].

However, its performance might be impacted if the dataset does not follow the feature independence assumption. Naive Bayes can nevertheless be very effective in spite of this drawback, particularly when it comes to text classification tasks like sentiment analysis and spam detection. Conversely, Random Forest functions as a decision tree-based ensemble learning method. It builds several decision trees during the training phase, then aggregates or combines the predictions from each tree to produce the final predictions. Every decision tree in the Random Forest is trained using a randomized subset of features and data points in order to combat overfitting and strengthen generalization. By using this method, Random Forest becomes resistant to noise and anomalies in the dataset, making it useful for tasks involving both regression and classification. It also provides feature important scores, which expedites the feature selection procedure as shown in Table 1.

The analysis using graphs for the Naïve Bayes, SVM, Random Forest, and ANN machine learning algorithms is shown in Figure 2. The accuracy, precision, and recall values are displayed on the graph. The model selection framework uses a number of different techniques to assess performance optimality. According to the graph, the RF is maximizing among them.

Table 1. Performance of classifiers.

Algorithm	"Accuracy"	"Precision"	"Recall"	"F1_Score"
RF	0.94	0.93	0.94	0.94
ANN	0.84	0.83	0.83	0.84
SVM	0.82	0.79	0.82	0.82
NB	0.75	0.73	0.75	0.75

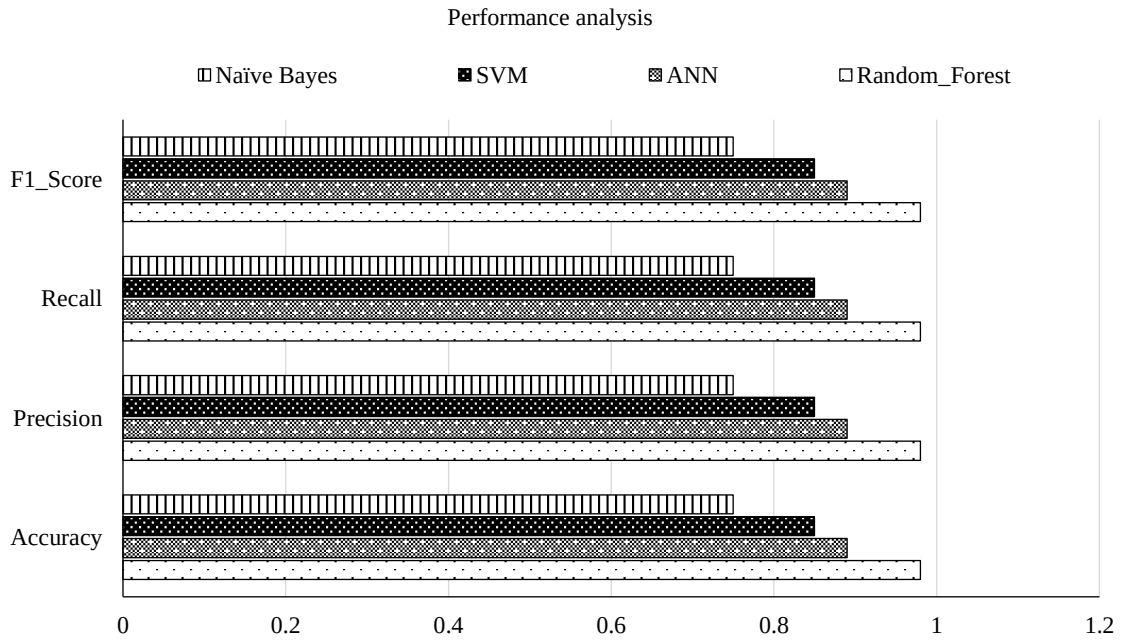

Figure 2. Classification result analysis.

CONCLUSION

Information security management is an essential component of creating secure networks and applications in the information technology industry. Our goal is to develop an Artificial Neural Network (ANN)-based Botnet Detection System (BDS) to achieve this. The goal of the BDS is to gather network data and use an ANN for analysis. We want to use the KDD_Cup99 dataset to train our classifier with the most pertinent features in order to accomplish this, allowing us to categories attacks and evaluate the algorithms' precision, accuracy, and detection rate. As a result, we are always working towards the goal of successfully categorizing botnets and the classification algorithm RF is obtaining maximum accuracy in the context of classification.

REFERENCES

1. Alhakami W. Alerts clustering for intrusion detection systems: overview and machine learning perspectives. Int J Adv Comput Sci Appl. 2019; 10(5): 573–582.
2. Almseidin M, Alzubi M, Kovacs S, Alkasassbeh M. Evaluation of machine learning algorithms for intrusion detection system. In 2017 IEEE 15th international symposium on intelligent systems and informatics (SISY). 2017 Sep 14; 000277–000282.
3. Iraqi O, El Bakkali H. Application-Level Unsupervised Outlier-Based Intrusion Detection and Prevention. Secur Commun Netw. 2019; 2019(1): 8368473.
4. Lai Y, Zhang J, Liu Z. Industrial anomaly detection and attack classification method based on convolutional neural network. Secur Commun Netw. 2019; 2019(1): 8124254.
5. Min E, Long J, Liu Q, Cui J, Chen W. TR-IDS: Anomaly-based intrusion detection through text-convolutional neural network and random forest. Secur Commun Netw. 2018; 2018(1): 4943509.
6. Yan B, Han G. LA-GRU: Building Combined Intrusion Detection Model Based on Imbalanced Learning and Gated Recurrent Unit Neural Network. Secur Commun Netw. 2018; 2018(1): 6026878.
7. Woodiss-Field A, Johnstone MN, Haskell-Dowland P. Examination of Traditional Botnet Detection on IoT-Based Bots. Sensors. 2024 Feb 5; 24(3): 1027.
8. Lagraa S, Husák M, Seba H, Vuppala S, State R, Ouedraogo M. A review on graph-based approaches for network security monitoring and botnet detection. Int J Inf Secur. 2024 Feb; 23(1): 119–40.

9. Wu G, Wang X, Zhang J. PeerG: A P2P botnet detection method based on representation learning and graph contrastive learning. Comput Secur. 2024 May 1; 140: 103775.
10. Wei C, Xie G, Diao Z. A lightweight deep learning framework for botnet detecting at the IoT edge. Comput Secur. 2023 Jun 1; 129: 103195.
11. Moorthy RS, Nathiya N. Botnet detection using artificial intelligence. Procedia Comput Sci. 2023 Jan 1; 218: 1405–13.
12. Xing Y, Shu H, Zhao H, Li D, Guo L. Survey on botnet detection techniques: Classification, methods, and evaluation. Math Probl Eng. 2021; 2021(1): 6640499.
13. Shinan K, Alsubhi K, Alzahrani A, Ashraf MU. Machine learning-based botnet detection in software-defined network: A systematic review. Symmetry. 2021 May 12; 13(5): 866.
14. Shi WC, Sun HM. DeepBot: a time-based botnet detection with deep learning. Soft Comput. 2020 Nov; 24(21): 16605–16.
15. Ibrahim WN, Anuar S, Selamat A, Krejcar O, Crespo RG, Herrera-Viedma E, Fujita H. Multilayer framework for botnet detection using machine learning algorithms. IEEE Access. 2021 Feb 22; 9: 48753–68.
16. Mahesh B. Machine learning algorithms-a review. Int J Sci Res (IJSR). 2020 Jan; 9(1): 381–6.
17. Sarker IH. Machine learning: Algorithms, real-world applications and research directions. SN Comput Sci. 2021 May; 2(3): 160.
18. Saranya T, Sridevi S, Deisy C, Chung TD, Khan MA. Performance analysis of machine learning algorithms in intrusion detection system: A review. Procedia Comput Sci. 2020 Jan 1; 171: 1251–60.

