International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue: 5 Volume No.6 Aug-Sept – 2022
DOI:10.46647/ijetms.2022.v06i05.062 ISSN: 2581-4621

DETECTION OF BOTNET ATTACK IN INTERNET OF THINGS USING FEATURE SELECTION ENABLE MACHINELERANING TECHNIQUE

Kevin G Kuriakose[1], Dr. Susheel George Joseph[2], Rahul P K[3], Rony Monichen[4], Dheeraj Kumar S[5]
[1,3,4,5]MCA – Kristu Jyoti College of Management & Technology, Changanassery,Kerala
[2]Associate Professor – Department of Computer Application, Kristu Jyoti College of Management & Technology, Changanassery,Kerala

Abstract – The Internet of Things (IoT) is very rapidly developing with millions of devices that are usedin the smart home ,smart city ,and many other smart systems for education and so on . On the other side , attackers are mostly targeting these devices. After infecting malware attacks on these devices ,they become bots that are controlled by attackers ,and these will be targeted to these organization not only for stealing important information but also for breaking down of the network.
Although certain security measures have been created to guard against cyber-attacks, themajority of these systems are rule-based. Additionally, malware attackers may be able to avoid the formal rule-based detection. Theabsence of earlier detection strategies is thereforefilled by the machine learning-based detection scheme. The public IDS dataset N-baloT andmachine learning techniques like the CART algorithm provide the foundation of thesuggested detection architecture.
Machine learning models based on fourclassifiers are built :Naïve bayes ,K-nearest neighbour ,Support vector Machine ,and decision trees. Using 82,000 records from UNSW-NB15 dataset, the decision tree model has yieldedthe best overall result with 99,89% testing accuracy, 100% precision, 100% recall, and 100% F-score in detecting botnet attacks.

I. INTRODUCTION

More common household items are connectingto the internet as a result of the Internet of Things' (IoT) ongoing technological development [1]. This increases the number ofdevices that could join a botnet. The purpose of this paper is to identify botnet assaults usingmachine learning techniques.
Multiple internet-connected devices that mayhave been purposefully infected with malware by cybercriminals make up a botnet. Distributeddenial-of-service (DDoS) attacks, data theft, and device access are all possible with botnets.A malicious attack known as a "botnet attack" makes use of a network of linked computers to attack or take down a network, network device,website, or IT environment. It is committedsolely with the intention of interfering withregular business activities or lowering the targetsystem's level of overall service. Botnet identification and prevention would therefore be extremely important for computer security.
Various machine learning techniques can beused to identify and separate these botnet devices when more devices become candidatesto be botnet devices. This research intends to increase the accuracy of previous related work by detecting botnets or malicious traffic activityusing new machine learning techniques.
The essay is organised as follows. Section II contains a review of the literature on related research. Section III presents the suggested methodology. Section IV includes a descriptionof the experimental results. Recommendations for further development are made in Section V'sconclusion.

II.LITERATURE REVIEW

The effectiveness of utilising machine learningand deep learning to identify the growing number of botnet attacks has been demonstratedin numerous research in recent years.

Finding the main traits or characteristics of a botnet that can aid distinguish between an attackand regular traffic is another area of study. Theapplication and efficacy of machine learning inbotnet detection were discussed by Dong et al. in [2]. In order to distinguish botnet traffic fromregular traffic, researchers looked at the structure of botnets. When creating our machine learning model, these features can then be utilised to choose other features. One such technique was utilised by Vishwakarma et al. [3], who generated data from an IoT network and deployed a honeypot to entice attackers.

The various aspects of an assault, such as IP addresses, MAC addresses, packet sizes, etc., were then examined using this new data. Some research focuses on identifying the key traits orcharacteristics of a botnet that can aid differentiate between an attack and regular traffic. Machine learning's application andefficiency in botnet identification were discussed by Dong et al. [2]. In order to distinguish botnet traffic from other types of traffic, researchers looked at the structure of botnets. When creating our machine learning model, these attributes can then be used to choose features. A honeypot was utilised by Vishwakarma et al. [3] to entice attackers and generate data from an IoT network. The many aspects of an assault, such as IP addresses, MAC addresses, packet sizes, etc., were then examined using this brand-new data. Additionally, Guerra-Manzanares et al. [4] presentedThe Decision Tree (DT) Classifier is one of themost promising classifiers for P2P botnet identification. For botnet detection, Haq andSingh [5] employed a variety of classifiers in addition to clustering. Over 38,000 records of network traffic, including both attack and regular traffic, were included in the collection. The DT classifier in this study had the highest accuracy, at 90.2723 percent, followed by the Decision Tree classifier, at 87.7853 percent.Similar to this, Khan et al. [6] found that P2P botnets were challenging to identify since they possessed conventional characteristics of centralization and spread. A two-stage detection approach for P2P botnets was proposed by Khan et al. in 2013. Data flow count, port judgement, and DNS query made up the initial stage of the filtering process for non-P2P traffic. To cut down on the number of packets being evaluated, the second stage made use of session feature bases. Additionally, the traffic was being classified and identified using machine learning algorithms. The experiment was conducted using the CTU-dataset, which contains 13 different botnet samples.

To identify P2P botnet traffic, three primary ML algorithms based on session characteristics were used.

The decision tree and random forest classifiers are further efficient classifiers. Stevanovic and Pedersen [7] investigated how supervised machine learning can be used to detect botnets with high accuracy. First, they suggested a system for detecting botnets that makes use of flow-based traffic monitoring and supervised machine learning. The performance of eight of the most significant machine learning algorithms (MLAs) for categorising botnet traffic is then tested. Finally, they looked at howmuch traffic should be visible for a categorization to be successful. Either "limited"analysis, where time intervals and packet countsare restricted, or "batch" analysis, which monitors from the beginning to the end of the trace, were used for traffic analysis. The ISOP dataset, which contains malicious and non- malicious records, was used for the trials. The findings demonstrated that while the random forest classifier had the highest accuracy ofbotnet detection, the random tree classifier wasthought to be the best option since it had the bestbalance of accuracy and detection time. In a more recent work from 2018, Hoang et al. [8] suggested comparing machine learning techniques for botnet identification to anomaly-based botnet detection methods. Based on their DNS logs, six distinct characteristics of botnet domain traffic were chosen. Name-based criteria, such the meaningful length ratio, madeup the majority of the chosen attributes.Message-based features such as the quantity of source IPs, kinds, and A, AAAA, NS, and MX queries came next. Finally, features that dependon quantity, such the number of queriesexecuted daily and per hour. Three well-knownML classifiers were used to determine the malicious domains from the DNS traffic once characteristics were chosen. The classifiers utilised were NB, Adaboost, and Bagging. The outcomes demonstrated strong performance, with precision rates above 90% for all classifiers and barely discernible variations between them. These outcomes unequivocally show the effectiveness of identifying andstopping malicious botnet behaviours when their domain names are present in the stream. Jin et al. advise implementing this approach with bigger DNS logs for future studies.

Three machine learning techniques were examined by Garg et al. [10] for their capacity to distinguish between botnet and regular traffic. This was accomplished by picking important aspects of network traffic, combining them into various combinations, and then creating a variety of test cases for the algorithms. The NB, Nearest Neighbor (IBk), and J48 were the three algorithms that were put to the test. The detection accuracy of the J48 and IBk algorithms was greater than that of NB after testing each algorithm with each case separately, although with the constraints that J48 required a high training time and IBk required a high testing time. Despite these drawbacks, P2P botnets could be found thanks to high detection rates of more than 99 percent.

Three machine learning techniques were examined by Garg et al. [10] for their capacity to distinguish between botnet and regular traffic. This was accomplished by picking important aspects of network traffic, combining them into various combinations, and then creating a variety of test cases for the algorithms. The NB, Nearest Neighbor (IBk), and J48 were the three algorithms that were put to the test. The detection accuracy of the J48 and IBk algorithms was greater than that of NB after testing each algorithm with each case separately, although with the constraints that J48 required a high training time and IBk required a high testing time. Despite these drawbacks, P2P botnets could be found thanks to high detection rates of more than 99 percent.

KNN classifiers are frequently employed in thisfield of classification. A study on anomaly-based mobile botnet detection for Android malware was carried out in 2013 by Feizollah etal. [13]. K-NN, Multi-Layer Perceptron (MLP),DT, and SVM were among the five machine learning classifiers utilised in the study. In this study, they tested the classifiers using malwaredata samples from the Android Malware Genome Project. Three network characteristicswere used for this study: the number of GET/POST parameters, the TCP size, and the connection time. The outcome demonstratedthat the KNN, which has a true positive rate of up to 99.94% and a false positive rate of 0.6%, is the best classifier. A network using DT, a machine learning classifier, was proposed by Arif et al. in a more recent research published in2018 [14].

The ability to recognise botnet attacks has also been demonstrated through deep and unsupervised learning. Pour et almethod .'s [15]of detecting over 350 IoT botnets in darknet data used a multi-window convolution neural network with clustering.

In order to identify IoT botnets, Al Shorman et al. [16] suggested an unsupervised intelligent system built from SVM and Grey Wolf optimization. This model was able to decrease
the amount of characteristics used for detection and obtain a low detection time.

These research have shown that machine learning is quite useful and successful at detecting botnets. The primary contribution of this study is the development of a machine learning-based classification strategy for botnet identification.

II.PROPOSED METHODOLOGY

A.Dataset Description

For this effort, a number of datasets are available, including the UNSW-NB15 and Bot- IoT datasets. The Bot-IoT dataset was developed by setting up a botnet network in a controlled environment and watching the network traffic to collect any packets that were being delivered. It has around 72 million records and 42 features (27 integer, 13 float, and 2 string kinds). The dataset includes assaults like DDoS, DoS, OS Scan, and others that are categorised as malicious and regular traffic, respectively.

The other dataset, UNSW-NB15, has 2.5 million records and 43 characteristics (14 Float, 6 Strings, and 23 Integer types) that are divided into two categories: attack traffic and regular traffic. The attack category is further divided into subcategories. The dataset also includes records for fuzzers, backdoors, reconnaissance, and worm assaults in addition to DDoS and DoS attacks [17]. To construct the dataset, these records were first collected in pcap files and converted to CSV. UNSW Canberra has assembled both datasets and made them accessible to the public for research purposes [18]. Additionally, it is discovered that the UNSW-NB15 dataset is a more refined dataset with characteristics similar to those of the Bot-IoT dataset but a wider variety of harmful entries.
.The UNSW-NB15 dataset was used in this study because it is more thorough and has been deemed the best dataset for training by UNSW. In this study, 82,000 records were chosen at random and used. To categorise the training and testing models, the data was cleaned and processed.
Numbers were coded with categorical information such as the "proto", type of "service," "state," "sptks," "sload," and "attack cat." The data was split at random into a training set, which contains 80% of the data, and a testing set, which contains 20% of the data.

B.Feature selection and Dimensionality reduction

In order to minimise the dimensionality of the data for our model while maintaining the variance in the data, we employed feature selection and dimensionality reduction. Principal Component Analysis is one technique for dimensionality reduction (PCA). PCA is a data transformation technique that places the data into a new feature space where the first coordinate of the new space (known as the first principal component) represents the majority of the variance in the data, the second most variance on the second principle component, and so on. In this study, the characteristics with the highest variance are chosen to provide the classifiers' input. By following this procedure, we make sure that only pertinent features are chosen, enhancing the computational effectiveness and simplicity of the machine learning models. Additionally, by using all the features, this technique has the potential to minimise any overfitting that might develop.

C.Classification Algorithms

In this work, a number of classifiers were employed and assessed. This work made use of the Python Scikit-learn module.
Using the real and predicted labels from the model, the confusion matrix was constructed to calculate precision, recall, and F-Measure for the evaluation. These classifiers were employed in this study:
1)Naïve Bayes (NB) with Gaussian probabilities-
a probabilistic classifier that makes use of the Bayes theorem and presumes conditional independence among the various dataset features. Based on the training set, NB determines the class probability.
2)K-Nearest Neighbor (k-NN) –
a non-parametric technique for regression and classification. The majority of the test sample's k nearest neighbours are used by the model to determine the test sample's class in order to predict it.
3)Support Vector Machine (SVM) with nonlinearkernel, namely radial basis function (RBF)
establishes a decision boundary using samples from several classes. The chosen kernel function, as well as important hyperparameters like C, which regulates the tradeoff between the decision boundary's smoothness and classification accuracy, and gamma, which specifies the impact of the data points' distribution on the decision boundary's shape, are used to create the decision boundary's shape.
4)Decision Tree (DT)-
A classification model that resembles a tree, with each node specifying a test on a single feature and each branch leading down from that node denoting one of the possible values for that feature.

Before applying these classifiers, the dataset was randomly split into training and testing datasets The training data was used to train the classifiers.The classifiers were then assessed using the testing dataset by predicting the labels .Each classifier was evaluated and compared in accordance with what was covered in Section IV.]

IV.DISCUSSION OF RESULTS-

DT, SVM, KNN, and NB classifiers have all been used, and the outcomes of each classifier model have been examined.We examined the accuracy, precision, recall, F-measure, and confusion matrices for the prediction test and training sets for each classifier.The outcome is displayed in the table below.

A.Principal Component Analysis (PCA)- To decrease the number of dimensions in a dataset while maintaining its variance, PCA was used to the dataset.Table II displays the PCA analysis results for the dataset.The optimum parameters that produced the maximum accuracy were noted after a simulation with variable PCA parameters was done.The table shows that there were 43 components selected for DT and NB.The components utilised for k-NN and SVM with RBF kernel were both 20.The best number of components produced the maximum accuracy for DT and SVM with RBF kerned.

B .Feature Selection - In this study, the best features were chosen in order to decrease the dataset's dimensionality and eliminate the impact of training the classifiers with irrelevant features.The level of correlation between the features and the output labels was assessed using multiple runs of Chi-squared and ANOVA Fvalue.In terms of accuracy, it was discovered that Chi-squared criteria was the most appropriate method for ranking the attributes.Only 14 features ensured achieving the best accuracy when using SVM with RBF.However, 29 attributes allowed DT to reach its peak level of accuracy.

Model Selection and Classification: In this study, we conducted grid search with a 5-fold cross validation on all the models using the best features identified in each classifier to obtain the best set of hyperparameters for each model and to guarantee its effectiveness.Table IV displays the results in summary.The optimal hyper parameter values for the NB classifier were discovered to be 1e-20 for "var smoothing," which produced the best results.The model performs reasonably well when compared to other models, where precision, accuracy, recall, and f-score all exceed 96 percent and training and testing accuracy are, respectively, 97 and 96 percent.As a result of producing some false positives and false negatives, the model creates an unusually high number of erroneous classifications in the confusion matrix. However, this model demonstrates to be a useful approach for categorising botnet attacks.

Similar to this, it was discovered that the optimal hyper parameter for the number of neighbours, "K neighbors," is 3. This was done using grid search on KNN.

It is apparent that when compared to the other models, the classifier's performance results were typically the worst.Precision, accuracy, recall, and f-score are all around 81 percent for this model, whereas training and testing accuracies were reported to be 91and 83 percent, respectively.According to the confusion matrix, there are a lot of false positives and false negatives classifications

The dataset records for the SVM with RBF kernel were limited to 10,000 because the SVM classifier is computationally expensive.The optimal hyperparameter values were C: 10 and gamma: 0.0001.The training and testing accuracy rates were 100% and 98.5%, respectively, with acceptable precision, accuracy, recall, and f-score values utilising this model.Finally, we determined that the ideal hyper parameters for DT are a maximum depth of 100, a maximum number of features at 10, a maximum number of leaf nodes at
none, and a minimum sample leaf of 10.Due to its usage of the whole dataset and best performance

values, this model outperforms other classifiers
zero incorrect classifications in the confusion matrix. The training and testing accuracy are averaged at 99.9% and 99.8%, respectively.
The effectiveness of the other machine learning models put forth in the literature was enhanced by this work.It showed a 9.8 percent increase over the work suggested in [5] and a 1 percent gain in detection accuracy over the work at [4] and [12].

TABLE IV SUMMARY OF PERFORMANCERESULTS FOR ALL THE COMPARED CLASSIFIERS

	Evaluation	Classifier - 5-foldvalidation			
		DT	NB Gaussian	SVM-rbf	KNN
Accuracy	Training accuracy	99.91 %	97.10 %	100%	91%
	Testing accuracy	99.89 %	96.90 %	98.80 %	83.00 %
Confusion Matrix	TP	7443	6986	22	6024
	FP	8	461	24	1346
	FN	10	53	0	1422
	TN	9006	8967	1954	7646
Precision	Normal traffic	100%	99%	100%	81%
	Attack traffic	100%	95%	99%	85%
	Macro avg	100%	97%	99%	83%
	Weighted avg	100%	97%	99%	83%
Recall	Normal traffic	100%	94%	48%	81%
	Attack traffic	100%	99%	100%	84%

V.CONCLUSION AND FUTURE WORK

The detection of botnet or malicious traffic behaviour using cutting-edge machine learning algorithms was suggested in this research.
In this paper, Naive Bayes, K-Nearest Neighbor, Support Vector Machine, and Decision Trees were used as classifiers.The experimental results showedthat the decision tree model performed somewhat better than the models that had previously beendiscussed in the literature under evaluation as well as better than the other classifier models.
This model has the potential to identify variousbotnet assaults and other types of harmful network

activities. techniques utilised in this paper can be contrasted with unsupervised learning techniques like clustering.
Further refining of these results can be done by looking into alternative feature selection techniques.Last but not least, the machine learning model may be tested in a real-time, controlled setting to precisely gauge how well it performs.

REFERENCES

[1] S. Ranger, "What is the IoT? Everything youneed to know about the Internet of Things right now
| ZDNet," ZDNet, 2020. [Online]. Available: https://www.zdnet.com/article/what-is-the-internet-ofthings-everything-you-need-to-know-about-the- iot-right-now/.
[2]X. Dong, J. Hu and Y. Cui, "Overview of Botnet Detection Based on Machine Learning," International Conference on Mechanical, Controland Computer Engineering, Huhhot, pp. 476-479, 2018.
[3] R. Vishwakarma and A. Jain, "A Honeypot withMachine Learning based Detection Framework for defending IoT based Botnet DDoS Attacks," International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, pp. 1019-1024, 2019.
[4] A. Guerra-Manzanares, H. Bahsi and S. Nõmm, "Hybrid Feature Selection Models for Machine Learning Based Botnet Detection in IoT Networks," International Conference on Cyberworlds (CW), Kyoto, Japan, pp. 324-327, 2019.
[5] S. Haq and Y. Singh, "Botnet Detection using Machine Learning," International Conference on Parallel, Distributed and Grid Computing (PDGC), India, pp. 240-245, 2018. [6] R. Khan, R. Kumar, M. Alazab and X. Zhang, "A Hybrid Technique To Detect Botnets, Based on P2P Traffic Similarity," Cybersecurity and Cyberforensics Conference (CCC), Melbourne, Australia, pp. 136-142, 2019.
[6] M. Stevanovic and J. Pedersen, "An efficient flow-based botnet detection using supervised machine learning," International Conference on Computing, Networking and Communications, HI, pp. 797-801, 2014.
[7] X. Hoang and Q. Nguyen, "Botnet Detection Based On Machine Learning Techniques Using DNS Query Data," Future Internet, vol. 10, no. 5, p. 43, May 2018
[8] M. Stevanovic and J. Pedersen,"An efficient flow-based botnet detection using supervised machine learning," International Conference on Computing, Networking , HI, pp. 797-801, 2014.
[9] S. Garg, A. Singh, A. Sarje and S. Peddoju, "Behaviour analysis of machine learning algorithms for detecting P2P botnets," International Conference on Advanced Computing Technologies, pp. 1-4, 2013.
[10] S. Saad et al., "Detecting P2P botnets through network behavior analysis and machine learning," International Conference on Privacy, Security and Trust, Montreal, QC, pp. 174-180, 2011.
[11] K.-C. Lin, S.-Y. Chen, and J. C. Hung, "Botnet Detection Using Support Vector Machines with Artificial Fish Swarm Algorithm," Journal of Applied Mathematics, vol. 2014, Article ID 986428, 2014.
[12] A. Feizollah, N. B. Anuar, R. Salleh, F. Amalina, R. Ma'arof, and S. Shamshirband, "A Study Of Machine Learning Classifiers for Anomaly-Based Mobile Botnet Detec