

Performance evaluation of botnet detection using machine learning techniques

Sneha Padhiar, Ritesh Patel

U and P U. Patel Department of Computer Engineering, CHARUSAT University, Gujarat, India

Article Info

Article history:

Received Feb 3, 2023

Revised Jul 11, 2023

Accepted Jul 17, 2023

Keywords:

Command and control (C and C server)

CTU-13

Machine learning

Performance evaluation

Traffic detection

ABSTRACT

Cybersecurity is seriously threatened by Botnets, which are controlled networks of compromised computers. The evolving techniques used by botnet operators make it difficult for traditional methods of botnet identification to stay up. Machine learning has become increasingly effective in recent years as a means of identifying and reducing these hazards. The CTU-13 dataset, a frequently used dataset in the field of cybersecurity, is used in this study to offer a machine learning-based method for botnet detection. The suggested methodology makes use of the CTU-13, which is made up of actual network traffic data that was recorded in a network environment that had been attacked by a botnet. The dataset is used to train a variety of machine learning algorithms to categorize network traffic as botnet-related/benign, including decision tree, regression model, naïve Bayes, and neural network model. We employ a number of criteria, such as accuracy, precision, and sensitivity, to measure how well each model performs in categorizing both known and unidentified botnet traffic patterns. Results from experiments show how well the machine learning based approach detects botnet with accuracy. It is potential for use in actual world is demonstrated by the suggested system's high detection rates and low false positive rates.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Sneha Padhiar

U and P U. Patel Department of Computer Engineering, CHARUSAT University

Gujarat-388421, India

Email: snehapadhiar.ce@charusat.ac.in

1. INTRODUCTION

An ever-changing threat scenario is accompanied by an increasing complexity in internet architecture. Hackers seek to discover ways to take advantage of weaknesses that may occur in a range of contexts, including devices, data, applications, people, and places. Botnets are a serious concern. There are three components of a botnet: the botmaster, the infected machine, and the administrative server (C and C server). It takes two steps for a botnet to communicate: first, a botmaster sends instructions to the botnet via remote link or directly to the bots. As a result of this, the controlled bots are able to carry out malicious actions after receiving malicious commands [1]. The threat of botnets compromising the fundamental principles of confidentiality, integrity, and availability is becoming increasingly clear as botnets pose an increasing threat to network security. It is especially important to note that distributed denial of service (DDoS) attacks can be launched using botnets that have a negative impact on the availability and performance of a network [2].

In general, botnet detection is done from two different angles: host-based and network-based. An abnormal use of computation resources can be identifies using the first technique. As an example, it monitors abnormally high central processing unit (CPU) usage and memory consumption. An analysis of the bot's

network and traffic conditions would be carried out using the later technique [3]. Its advantage is that it can be applied even when communication is encrypted. In contrast, this method is more time-consuming and requires continuous monitoring of all host's resource utilization. The two types of network-based techniques are signature based and anomaly based. An approach based on signature is used to apply deep packet inspection (DPI) to internet protocol (IP) packets. Low false positive rates are a benefit of it. Identifying known botnets is its primary use [4]. The drawback is in concern to identify new patterns of attacks; signatures must regularly be updated. In addition, encryption techniques can conceal the signatures. Anomaly-based techniques can be used to find anomalies based on variables like packet payload size and bot activity. It is more challenging to identify botnet attacks as time goes on due to the frequent changes in botnet behavior [5], [6]. Due to their abilities to detect anomalous traffic patterns, machine learning techniques have become increasingly popular in anomaly-based methods. However, anomaly-based detection generally caused many detection errors due to high false positive rate. Additionally, one significant drawback of traditional machine learning methods is that they demand a lot of work and depend on a time-consuming feature engineering procedure. These limitations have led to the recent proposal of and growing interest in machine learning methods based on neural networks for applications in network security. With machine learning, it is possible to independently choose the most appropriate features out of all the features [7]–[12]. As a result, machine learning approaches are excellent for handling data sets with a variety of properties. In this paper, we use the CTU-13 dataset, a real botnet traffic dataset created in the Czech Republic at the CTU University in 2011, to provide a study of four machine learning models, including decision tree, regression model, naive Bayes model, and neural network models [13], [14]. An investigation of the impact of having uneven traffic statistics is another component of this paper (i.e., a significant disparity between the volume of benign and botnet traffic). Accuracy, precision, and sensitivity are taken into account as performance indicators [15], [16]. We conclude by comparing the effectiveness of our machine learning based models to that of common machine learning methods [17]–[20].

The remaining of the paper is organized as follows. In section 2, we go over earlier research on machine learning-based botnet identification. We go over the performance indicators and machine learning models in section 3. The results of the performance evaluation are provided in section 4 along with an analysis. The paper's conclusion in section 5, summarizes our key findings and outlines concepts for more research.

2. RELATED WORK

We found several works on detection of botnets, DDoS attacks, network traffic, and other related topics in our literature survey. The majority of research is based on machine learning. Table 1 (see in appendix) provides a comparison of related work's survey results. Based on their chosen methodologies, datasets and outcomes, the author's contributions are summarized in the Table 1. Using literature research as a baseline Four distinct machine learning models are used, and they are carefully compared, such as decision trees, regression models, naïve bayes models and neural network models. Table 1 summarizes the results of a literature survey that covers these model's previous use for anomaly detection and how they obtained promising results.

3. PROPOSED METHOD

A botnet attack proceeds through several stages shown in Figure 1, such as scanning for malware, injecting malware, connecting to a botnet, executing commands, and maintaining and upgrading the botnet. A botnet attack begins with scanning, which is the first phase of the attack. During scanning, the proposed methodology stops attackers from progressing further in their attacks. As a result, the suggested methodology distinguishes between botnet attacks that are detected by both inbound and outbound DDoS attacks and those that are prevented by detecting scanning activity.

As illustrated in Figure 2, the proposed method successfully navigates the key phases of attack detection and scanning. In order to train the machine learning models, we used the CTU-13 scanning and DDoS attack traffic data that was acquired (in *.pcap* format) in the first step. The features were then retrieve from these network packet traces (*.pcap* files) and stored in *.csv* files in the subsequent stage [21]. To further distinguish between different types of traffic, we added labels to the dataset, such as "normal" for routine traffic, "scan" for scanning traffic, and "DDoS" for DDoS attack traffic. In the third phase, we applied the logistic regression (LR) feature selection technique to enhance the performance of the machine learning model [21]. LR feature selection was chosen over feature selection techniques because of it is efficient performance, simplicity, and low complexity. In the final step, we train machine learning (ML) models and validate their performance against a real-time attack scenario.

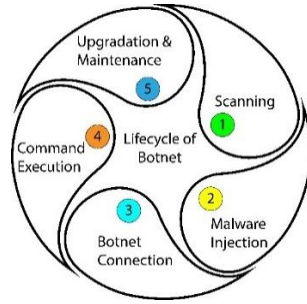


Figure 1. Botnet life cycle

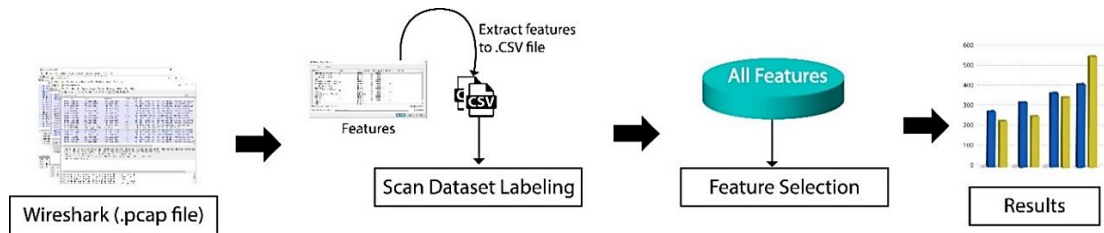


Figure 2. Proposed approach to detect botnet attack

3.1. CTU-13 dataset

To evaluate the machine learning based models, we use the CTU-13 dataset produced in the Czech Republic in 2011 at CTU college. The CTU-13 dataset, which is organized in a grid, contains 13 instances from various types of botnets. Even though it was first developed in 2011, it is still relevant for current botnet research because every situation includes explicit traffic results from a specific botnet and for different attack types. Table 2 lists the types and numbers of bots for each scenario as well as the different forms of botnet attack. For each situation, we list the botnet sizes and optimal traffic flow rates in Tables 2 and 3.

3.2. Feature selection and dataset pre-processing

The CTU-dataset is used to extract the following features: type of service, direction, destination address, total packet, total bytes, source bytes, protocol, source address, source port. Additionally, we include additional data to help us distinguish between legitimate traffic and botnet traffic, such as average bytes per second and packets per second. An abnormal value of these features indicates the entry of botnet traffic into the network. The next step involves splitting the data into training data, validation data, and testing data. Machine learning models are trained using training data and tested using testing data to determine how well they predict real-world traffic patterns. In each learning round (that is, after training has been completed), we verify whether the model was trained correctly using validation data. The binary classification process is further enhanced by adding labels “0” and “1” for botnet traffic and benign, respectively.

Table 2. Botnet attack scenarios of the CTU-13 dataset

Scenario	Attack type	Bot Type	Number of Bots
1	IRC, SPAM, CF	Neris	1
2	IRC, SPAM, CF	Neris	1
3	IRC, PS	Rbot	1
4	IRC, DDoS	Rbot	1
5	SPAM, PS, HTTP	Virut	1
6	PS	Menti	1
7	HTTP	Sogou	1
8	PS	Murlo	1
9	IRC, SPAM, CF, PS	Neris	10
10	IRC, DDoS	Rbot	10
11	IRC, DDoS	Rbot	3
12	P2P	NSIS.ay	3
13	SPAM, PS, HTTP	Virut	1

IRC=internet relay chat, CF=click fraud, PS=Port Scan, HTTP=hypertext transfer protocol, P2P=Peer-to-Peer

Table 3. Number of traffic flows for each scenario of the CTU-13 dataset

Scenario	Number of botnet flows	Number of benign flows
1	2693	8839
2	14362	5267
3	24	27433
4	1931	23731
5	901	4679
6	4630	7494
7	63	1677
8	1520	36625
9	8686	16690
10	74907	13052
11	8164	2718
12	2168	7628
13	23779	13199

3.3. Machine learning-based botnet detection

This section discusses some background knowledge of the four different machine learning models name decision tree, naïve Bayes, regression model and neural network model, which we used to detect botnets malicious activity and adapt them. This machine learning algorithm has the ability to detect the activity of Botnet with good accuracy.

- a. Decision tree (DT): This model is built on a tree structure called a decision tree, where each node represents a test on a single feature and all branches that descend from that node indicate potential values for that feature. Training and testing datasets were randomly divided into these classes to be applied to the dataset. After classifiers were trained, they were then tested against the testing dataset by predicting label values [21].
- b. Naïve Bayes (NB) with Gaussian probabilities: The Bayes theorem is used to calculate the probabilities used in the classifier and this assumes conditional independence. As a result of training the class probability, NB produces an estimate of the class probability [21].
- c. Regression model: One or more independent variables are modeled by regression analysis to predict the relationship between the dependent (target) variable and the independent (predictor). When other independent variables are held constant, regression analysis reveals how the level of the dependent variable changes in relation to an independent variable [21].
- d. Neural network model: Artificial neural network (ANNs) are a form of deep learning algorithms and a subset of machine learning. Each ANN consists of a layer consisting of an input layer, a hidden layer, and an output layer. A subset of machine learning, ANNs are a type of deep learning algorithms. An input layer, a hidden layer and an output layer make up each layer of an ANN. It is made up of nodes, or artificial neurons, which are connected to each other and have a threshold and weight associated with each node. A node is activated if its output exceeds the specified threshold value and data is transmitted to the following layer of the network. Any other scenario prevents data from moving to the following tier of the network [21], [22].

4. PERFORMANCE EVALUATION RESULTS AND ANALYSIS

4.1. Evaluation metrics

The measures used to evaluate ID's performance are described in this section. The performance of four machine learning models for detecting botnet attacks is assessed using the metrics accuracy and false alarm rate. The confusion matrix entries used to determine the performance measures are displayed in Table 4.

Table 4. Confusion matrix

		Predicated class	
		Attack	Normal
Actual Class	Attack	True positive	False negative
	Normal	False positive	True negative

Where, true positive (TP) refers to situations where the classifier categories an attack accurately. False negative (FN) refers to situation in which the classifier incorrectly labels an attack as normal. False positive (FP) refers to situations where the classifier incorrectly labels a typical occurrence as an attack. True negative (TN) identifies situations where the classifier detects typical occurrences accurately. A number of other evaluation metrics are also used by researchers, including recall, true negative, accuracy, precision and false alarm rate. False alarm rate (FAR): the ration of samples that were mistakenly forecasted as attacks to all other sample is known as the false positive rate. $FAR = FP / (TN + FP)$. Accuracy: the categorization accuracy is related to the metric accuracy. It is calculated by dividing the number of input samples by the proportion of accurate predictions. $Accuracy = TP + TN / (TP + TN + FP + FN)$.

4.2. Results and discussion

CTU-13 scenarios are divided into training, validation and testing datasets in order to determine how well our models classify traffic from botnet attacks. Using the data from scenarios 3, 4, 5, 7, 10, 11, 12, and 13 the author recommends partitioning it as follows: training and validation data are used for scenarios 3, 4, 5, 7, and 11 while testing data are used for scenarios 1, 2, 6, 8, and 9. In terms of training and validation of data, the split ratio remains 80:20. Table 5, Figures 3 and 4 illustrates the results of the performance evaluation. The classifiers we used performed very well when it came to classifying botnet traffic flows. convolution neural network (CNN) and decision tree (DT) are capable of correctly identifying on and averages around 98% and 99% of traffic flows, respectively.

Table 5. Scenario wise result and findings

Scenario wise (accuracy and FAR) %		Classification model			
		DT	RM	NB	CNN
1	ACC	99.95	99.90	96.25	99.97
	FAR	0.05	0.1	3.75	0.03
2	ACC	100	100	100	100
	FAR	0	0	0	0
6	ACC	99.87	99.50	70.52	99.57
	FAR	0.03	0.5	29.48	0.43
8	ACC	99.95	99.90	71.90	99.95
	FAR	0.05	0.1	28.1	0.05
9	ACC	99.95	97.24	78.75	98.24
	FAR	0.05	2.76	21.25	1.76
10	ACC	99.68	98.79	65.37	95.79
	FAR	0.32	1.21	34.63	4.21

DT=decision tree, RM=regression model, NB=naïve Bayes, CNN=neural network model

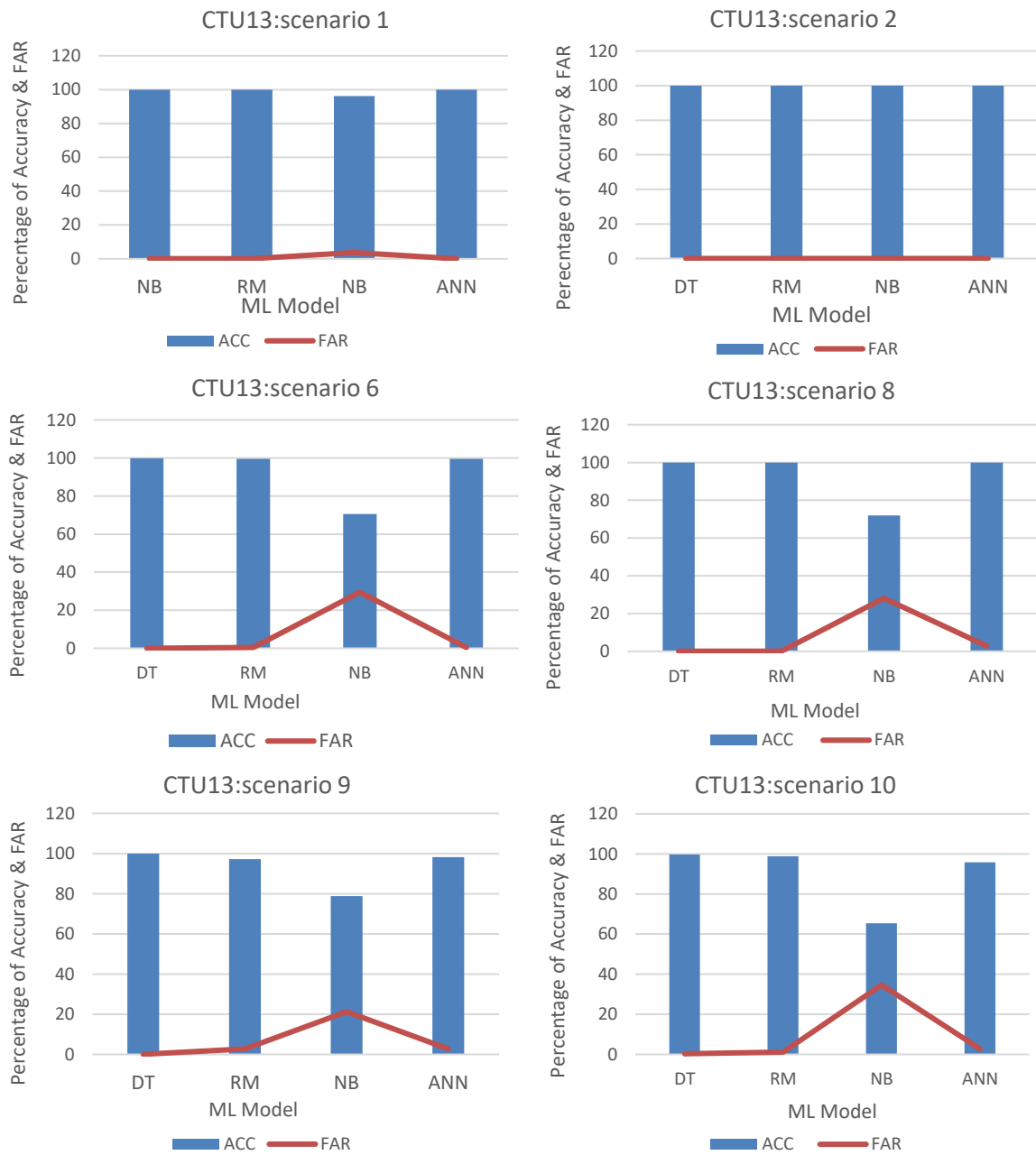


Figure 3. Scenario wise result and findings

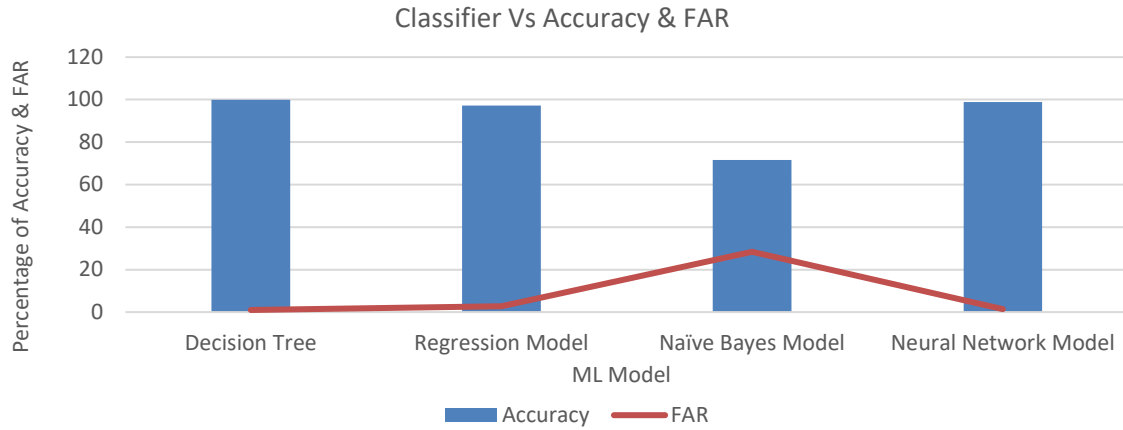


Figure 4. Classifier V/S accuracy and false alarm rate

4.3. Comparison with machine learning techniques

In this subsection, we contrast the effectiveness of our machine learning based methods with those of other researcher’s previously studied machine learning methods. Only the findings of earlier research using the CTU-13 dataset are presented here. Using samples from the CTU-13 dataset, Chen *et al.* [23] assessed the effectiveness of the J48, hybrid classification and clustering, tree classifier-means clustering. Their findings indicate that 90.2726% accuracy is maximum achievable. For the J48 tree classifier, it is obtained. Popular machine learning methods like support vector, logistic regression, random forests, and gradient boosting were used by Apruzzese *et al.* [24]. They used 2/3 of the CTU-13 dataset to train the algorithms and 1/3 of the dataset to evaluate them (randomly chosen). According to their analysis, the neural network and decision tree-based method works the best. The CTU-13 dataset’s scenario was used by the authors to further assess the method of machine learning. For each CTU-13 scenario, we compared our model to that of the prior authors, and the corresponding results are displayed in Figure 5 and Table 6.

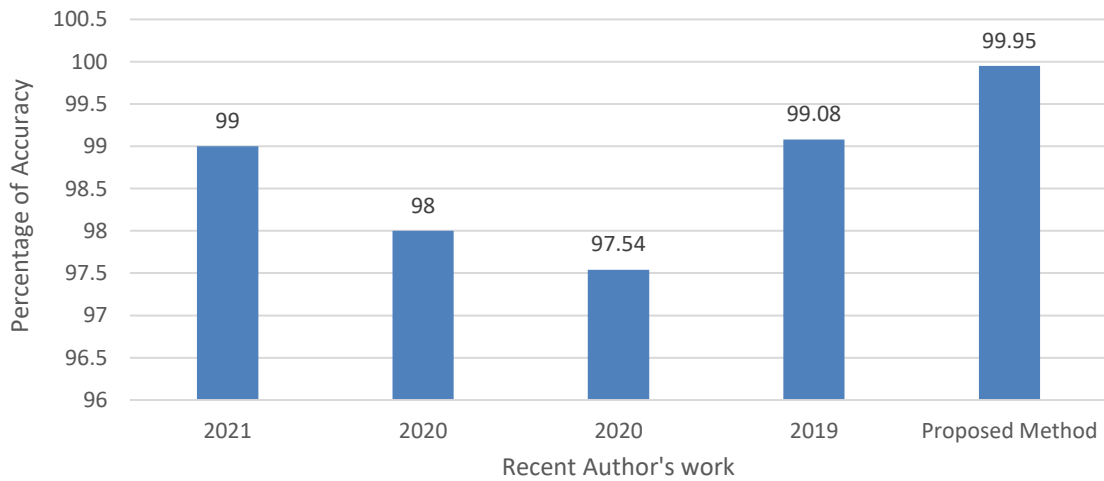


Figure 5. Comparison with existing model

Table 6. Comparison with existing model

Authors	Dataset	Method	Accuracy
[20]/2021	CTU13	KNN (10-cross fold)	99%
[25]/2020	CTU13	KNN	98%
[26]/2020	CTU13	Decision Tree	97.54%
[27]/2019	CTU13	Deep learning	99.08%
Proposed method	CTU13	Decision tree, Regression model, Naive Bayes, Neural network model	99.95%

5. CONCLUSION

This paper evaluates four machine learning models for detection of botnet attack using the dataset CTU-13, which is labelled data that constitutes real flow(s) of traffic, including decision tree, regression model, naïve Bayes and neural network models. In addition to the classification of traffic flows, we also explore the impact of inconsistent data and the impact of known and unknown botnet attacks. Among our models for predicting botnet traffic flows, the CNN and decision tree models produce the best results. An accuracy rate of 98.30% and 99.95% can be achieved by it. Therefore, 99% of benign traffic flows can be identified correctly with the model. In our future work, a real-time botnet attack detection system will be developed by combining the deep learning models and newer deep reinforcement learning techniques.

APPENDIX

Table 1. Summary of botnet detection based on ML




Ref/Year	Contribution	Protocol	Methods	Dataset	Result
[20]/2021	Analyzing host and network traffic to identify hybrid botnets.	HTTP IRC P2P	DT NB	ISCX CTU-13	99% Accuracy
[28]/2022	In order to identify botnets, DL was employed to study the behaviors of the traffic generated by packet on a network.	HTTP IRC	LSTM_R NN	Live data	98.36% Accuracy
[29]/2020	Use three modules to signal P2P botnet detection using machine learning: Extraction, selection, and classification of features.	P2P	DT J48	ISOT (2012) botnet (2014)	98.90% Accuracy
[30]/2020	Deep Learning-based P2P botnet detection	P2P	GNN	CAIDA	98.05% Accuracy
[31]/2020	Framework for IoT network botnet detection using sequential ML based detection architecture.	IRC HTTP P2P	ANN J48 DT NB	N-IoT	99% Accuracy
[25]/2020	Investigation of DDoS botnet detection methods based on machine learning.	IRC HTTP	DT ANN NB	KDD99	98.08% Accuracy
[26]/2020	Two-stage traffic classification is the foundation of a novel approach.	P2P	DT NB ANN	CTU 13	94.4% Accuracy
[32]/2020	Scalable deep learning-based botnet identification is addressed by DBD.	IRC HTTP	CNN_LS TM	Internal Network	97.80% Accuracy
[33]/2019	P2P traffic is classified as regular or botnet using a multi-layer method based on ML classifiers on network features.	P2P	DT, KNN	CTU, ISOT	98.7% Accuracy
[34]/2019	flow-based ML ensembles for detection of botnet	IRC HTTP P2P	GNB, NN, DT	CTU-13	0.99 F1 score
[27]/2019	Botnet identification based on network traffic analytics features and machine learning	IRC HTTP	J48 SVM	Real botnet sample collected in their laboratory	HTTP (accuracy 80%), IRC (accuracy 95%) False positive rate HTTP (0.05% FPR), IRC (0.025% FPR), FPR:0.23
[35]/2018	Botshark is a proposed deep learning-based botnet traffic analyzer.	IRC, P2P-	CNN	ISCX	
[21]/2022	Aims to find a botnet in environment of fast network	IRC, HTTP, P2P	RF	CTU	93.6% Accuracy

REFERENCES




- [1] B. Nugraha, A. Nambiar, and T. Bauschert, "Performance Evaluation of botnet detection using deep learning techniques," in *2020 11th International Conference on Network of the Future (NoF)*, Oct. 2020, pp. 141–149, doi: 10.1109/NoF50125.2020.9249198.
- [2] F. Hussain *et al.*, "A two-fold machine learning approach to prevent and detect IoT botnet attacks," *IEEE Access*, vol. 9, pp. 163412–163430, 2021, doi: 10.1109/ACCESS.2021.3131014.
- [3] C. Joshi, R. K. Ranjan, and V. Bharti, "A fuzzy logic based feature engineering approach for Botnet detection using ANN," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 9, pp. 6872–6882, Oct. 2022, doi: 10.1016/j.jksuci.2021.06.018.
- [4] A. McCarthy, E. Ghadafi, P. Andriotis, and P. Legg, "Functionality-preserving adversarial machine learning for robust classification in cybersecurity and intrusion detection domains: a survey," *Journal of Cybersecurity and Privacy*, vol. 2, no. 1, pp. 154–190, Mar. 2022, doi: 10.3390/jcp2010010.

- [5] R. H. Randhawa, N. Aslam, M. Alauthman, H. Rafiq, and F. Comeau, "Security hardening of botnet detectors using generative adversarial networks," *IEEE Access*, vol. 9, pp. 78276–78292, 2021, doi: 10.1109/ACCESS.2021.3083421.
- [6] Z. Lin, Y. Shi, and Z. Xue, "IDSGAN: generative adversarial networks for attack generation against intrusion detection," *arXiv preprint arXiv:1809.02077*, Sep. 2018.
- [7] A. A. Ahmed, W. A. Jabbar, A. S. Sadiq, and H. Patel, "Deep learning-based classification model for botnet attack detection," *Journal of Ambient Intelligence and Humanized Computing*, vol. 13, no. 7, pp. 3457–3466, Jul. 2022, doi: 10.1007/s12652-020-01848-9.
- [8] M. A. Ferrag, L. Maglaras, S. Moschoviannis, and H. Janicke, "Deep learning for cyber security intrusion detection: approaches, datasets, and comparative study," *Journal of Information Security and Applications*, vol. 50, Feb. 2020, doi: 10.1016/j.jisa.2019.102419.
- [9] P. T. Duy, L. K. Tien, N. H. Khoa, D. T. T. Hien, A. G.-T. Nguyen, and V.-H. Pham, "DIGFuPAS: Deceive IDS with GAN and function-preserving on adversarial samples in SDN-enabled networks," *Computers and Security*, vol. 109, Oct. 2021, doi: 10.1016/j.cose.2021.102367.
- [10] Q. Cheng, S. Zhou, Y. Shen, D. Kong, and C. Wu, "Packet-level adversarial network traffic crafting using sequence generative adversarial networks," *arXiv preprint arXiv:2103.04794*, Mar. 2021.
- [11] N. S. Alfaiz and S. M. Fati, "Enhanced credit card fraud detection model using machine learning," *Electronics*, vol. 11, no. 4, Feb. 2022, doi: 10.3390/electronics11040662.
- [12] J. Engelmann and S. Lessmann, "Conditional Wasserstein GAN-based oversampling of tabular data for imbalanced learning," *arXiv preprint arXiv:2008.09202*, Aug. 2020.
- [13] S. Huang and K. Lei, "IGAN-IDS: An imbalanced generative adversarial network towards intrusion detection system in ad-hoc networks," *Ad Hoc Networks*, vol. 105, Aug. 2020, doi: 10.1016/j.adhoc.2020.102177.
- [14] M. H. Shahriar, N. I. Haque, M. A. Rahman, and M. Alonso, "G-IDS: Generative adversarial networks assisted intrusion detection system," in *2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC)*, Jul. 2020, pp. 376–385, doi: 10.1109/COMPSAC48688.2020.0-218.
- [15] Z. Mao, Z. Fang, M. Li, and Y. Fan, "EvadeRL: evading PDF malware classifiers with deep reinforcement learning," *Security and Communication Networks*, vol. 2022, pp. 1–14, Apr. 2022, doi: 10.1155/2022/7218800.
- [16] R. H. Randhawa, N. Aslam, M. Alauthman, and H. Rafiq, "EVAGAN: evasion generative adversarial network for low data regimes," *arXiv preprint arXiv:2109.08026*, Sep. 2021.
- [17] T. Truong-Huu *et al.*, "An empirical study on unsupervised network anomaly detection using generative adversarial networks," in *Proceedings of the 1st ACM Workshop on Security and Privacy on Artificial Intelligence*, Oct. 2020, pp. 20–29, doi: 10.1145/3385003.3410924.
- [18] T.-D. Pham, T.-L. Ho, T. Truong-Huu, T.-D. Cao, and H.-L. Truong, "MAppGraph: Mobile-App classification on encrypted network traffic using deep graph convolution neural networks," in *Annual Computer Security Applications Conference*, Dec. 2021, pp. 1025–1038, doi: 10.1145/3485832.3485925.
- [19] S. Padihar and R. Patel, "Behaviour based botnet detection with traffic analysis and flow intervals at the host level," *Indonesian Journal of Electrical Engineering and Computer Science (IJECS)*, vol. 31, no. 1, pp. 350–358, Jul. 2023, doi: 10.11591/ijeecs.v31.i1.pp350-358.
- [20] W. N. H. Ibrahim *et al.*, "Multilayer framework for botnet detection using machine learning algorithms," *IEEE Access*, vol. 9, pp. 48753–48768, 2021, doi: 10.1109/ACCESS.2021.3060778.
- [21] M. S. Gadelrab, M. ElSheikh, M. A. Ghoneim, and M. Rashwan, "BotCap: machine learning approach for botnet detection based on statistical features," *International Journal of Communication Networks and Information Security (IJCNIS)*, vol. 10, no. 3, Apr. 2022, doi: 10.17762/ijcnis.v10i3.3624.
- [22] S. Homayoun, M. Ahmadzadeh, S. Hashemi, A. Dehghantanha, and R. Khayami, "BoTShark: A deep learning approach for botnet traffic detection," in *Advances in Information Security*, Springer International Publishing, 2018, pp. 137–153.
- [23] R. Chen, W. Niu, X. Zhang, Z. Zhuo, and F. Lv, "An effective conversation-based botnet detection method," *Mathematical Problems in Engineering*, pp. 1–9, 2017, doi: 10.1155/2017/4934082.
- [24] G. Apruzzese, M. Andreolini, M. Marchetti, A. Venturi, and M. Colajanni, "Deep reinforcement adversarial learning against botnet evasion attacks," *IEEE Transactions on Network and Service Management*, vol. 17, no. 4, pp. 1975–1987, Dec. 2020, doi: 10.1109/TNSM.2020.3031843.
- [25] J. Zhou, Z. Xu, A. M. Rush, and M. Yu, "Automating botnet detection with graph neural networks," *arXiv preprint arXiv:2003.06344*, Mar. 2020.
- [26] Y. N. Soe, Y. Feng, P. I. Santosa, R. Hartanto, and K. Sakurai, "Machine learning-based IoT-botnet attack detection with sequential architecture," *Sensors*, vol. 20, no. 16, Aug. 2020, doi: 10.3390/s20164372.
- [27] R. U. Khan, X. Zhang, R. Kumar, A. Sharif, N. A. Golilarz, and M. Alazab, "An adaptive multi-layer botnet detection technique using machine learning classifiers," *Applied Sciences*, vol. 9, no. 11, Jun. 2019, doi: 10.3390/app9112375.
- [28] S. Hosseini, A. E. Nezhad, and H. Seilani, "Botnet detection using negative selection algorithm, convolution neural network and classification methods," *Evolving Systems*, vol. 13, no. 1, pp. 101–115, Feb. 2022, doi: 10.1007/s12530-020-09362-1.
- [29] S. Almutairi, S. Mahfoudh, S. Almutairi, and J. S. Alowibdi, "Hybrid botnet detection based on host and network analysis," *Journal of Computer Networks and Communications*, vol. 2020, pp. 1–16, Jan. 2020, doi: 10.1155/2020/9024726.
- [30] W.-C. Shi and H.-M. Sun, "DeepBot: a time-based botnet detection with deep learning," *Soft Computing*, vol. 24, no. 21, pp. 16605–16616, Nov. 2020, doi: 10.1007/s00500-020-04963-z.
- [31] P. Gahelot and N. Dayal, "Flow based botnet traffic detection using machine learning," in *Proceedings of ICETIT 2019*, Springer International Publishing, 2020, pp. 418–426.
- [32] T. A. Tuan, H. V. Long, L. H. Son, R. Kumar, I. Priyadarshini, and N. T. K. Son, "Performance evaluation of Botnet DDoS attack detection using machine learning," *Evolutionary Intelligence*, vol. 13, no. 2, pp. 283–294, Jun. 2020, doi: 10.1007/s12065-019-00310-w.
- [33] R. U. Khan, R. Kumar, M. Alazab, and X. Zhang, "A hybrid technique to detect botnets, based on P2P traffic similarity," in *2019 Cybersecurity and Cyberforensics Conference (CCC)*, May 2019, pp. 136–142, doi: 10.1109/CCC.2019.00008.
- [34] R. Vinayakumar, K. P. Soman, P. Poornachandran, M. Alazab, and A. Jolfaei, "DBD: deep learning DGA-based botnet detection," in *Deep Learning Applications for Cyber Security*, Springer International Publishing, 2019, pp. 127–149.
- [35] S. Ryu and B. Yang, "A comparative study of machine learning algorithms and their ensembles for botnet detection," *Journal of Computer and Communications*, vol. 6, no. 5, pp. 119–129, 2018, doi: 10.4236/jcc.2018.65010.

BIOGRAPHIES OF AUTHORS

Sneha Padhiar    is an Assistant professor in U and P U. Patel Department of Computer Engineering, CHARUSAT University, Gujarat, India. She received her B.E.C.E. From Gujarat Technological University in 2014 and M.E.C.E. From Gujarat Technological University in 2016. Currently, she is pursuing doctoral course in Computer Engineering at CHARUSAT. Her major area of research includes information security and IoT. She can be contacted at email: snehapadhiar.ce@charusat.ac.in.



Ritesh Patel    is working as a Professor at U and P.U Patel Computer engineering department of CHARUSAT University of Science and Technology, Gujarat, India. He has received his Doctorate degree in 2017 from CHARUSAT University. His areas of interest include cloud computing, internet of things, communication and networking, computer architecture, software engineering, and cluster computing. He can be contacted at email: riteshpatel.ce@charusat.ac.in.