Performance evaluation of Botnet DDoS attack detection using machine learning

Article *in* Evolutionary Intelligence · June 2020

DOI: 10.1007/s12065-019-00310-w

6 authors, including:

Ishaani Priyadarshini
Cornell University
85 PUBLICATIONS 2,530 CITATIONS
SEE PROFILE

Raghvendra Kumar
Rohilkhand Medical College and Hospital
200 PUBLICATIONS 4,956 CITATIONS
SEE PROFILE

Evolutionary Intelligence
https://doi.org/10.1007/s12065-019-00310-w

SPECIAL ISSUE

Performance evaluation of Botnet DDoS attack detection using machine learning

Tong Anh Tuan[1] · Hoang Viet Long[1] · Le Hoang Son[2] · Raghvendra Kumar[3] · Ishaani Priyadarshini[4] · Nguyen Thi Kim Son[5,6]

Received: 31 August 2019 / Revised: 15 October 2019 / Accepted: 21 October 2019

Abstract

Botnet is regarded as one of the most sophisticated vulnerability threats nowadays. A large portion of network traffic is dominated by Botnets. Botnets are conglomeration of trade PCs (Bots) which are remotely controlled by their originator (BotMaster) under a Command and-Control (C&C) foundation. They are the keys to several Internet assaults like spams, Distributed Denial of Service Attacks (DDoS), rebate distortions, malwares and phishing. To over the problem of DDoS attack, various machine learning methods typically Support Vector Machine (SVM), Artificial Neural Network (ANN), Naïve Bayes (NB), Decision Tree (DT), and Unsupervised Learning (USML) (K-means, X-means etc.) were proposed. With the increasing popularity of Machine Learning in the field of Computer Security, it will be a remarkable accomplishment to carry out performance assessment of the machine learning methods given a common platform. This could assist developers in choosing a suitable method for their case studies and assist them in further research. This paper performed an experimental analysis of the machine learning methods for Botnet DDoS attack detection. The evaluation is done on the UNBS-NB 15 and KDD99 which are well-known publicity datasets for Botnet DDoS attack detection. Machine learning methods typically Support Vector Machine (SVM), Artificial Neural Network (ANN), Naïve Bayes (NB), Decision Tree (DT), and Unsupervised Learning (USML) are investigated for Accuracy, False Alarm Rate (FAR), Sensitivity, Specificity, False positive rate (FPR), AUC, and Matthews correlation coefficient (MCC) of datasets. Performance of KDD99 dataset has been experimentally shown to be better as compared to the UNBS-NB 15 dataset. This validation is significant in computer security and other related fields.

Keywords Botnet detection · Command and control channel · Distributed Denial of service attack · Machine learning · Unsupervised learning

✉ Nguyen Thi Kim Son
nguyenthikimson@tdtu.edu.vn

Tong Anh Tuan
tuanqb92@gmail.com

Hoang Viet Long
longhv08@gmail.com

Le Hoang Son
sonlh@vnu.edu.vn

Raghvendra Kumar
raghvendraagrawal7@gmail.com

Ishaani Priyadarshini
ishaani@udel.edu

[1] The People's Police University of Technology and Logistics, Thuận Thành, Bac Ninh, Vietnam

[2] VNU Information Technology Institute, Vietnam National University, Hanoi, Vietnam

[3] Department of Computer Science and Engineering, LNCT College, Bhopal, India

[4] University of Delaware, Newark, DE, USA

[5] Division of Computational Mathematics and Engineering, Institute for Computational Science, Ton Duc Thang University, Ho Chi Minh City, Vietnam

[6] Faculty of Mathematics and Statistics, Ton Duc Thang University, Ho Chi Minh City, Vietnam

Published online: 20 November 2019

1 Introduction

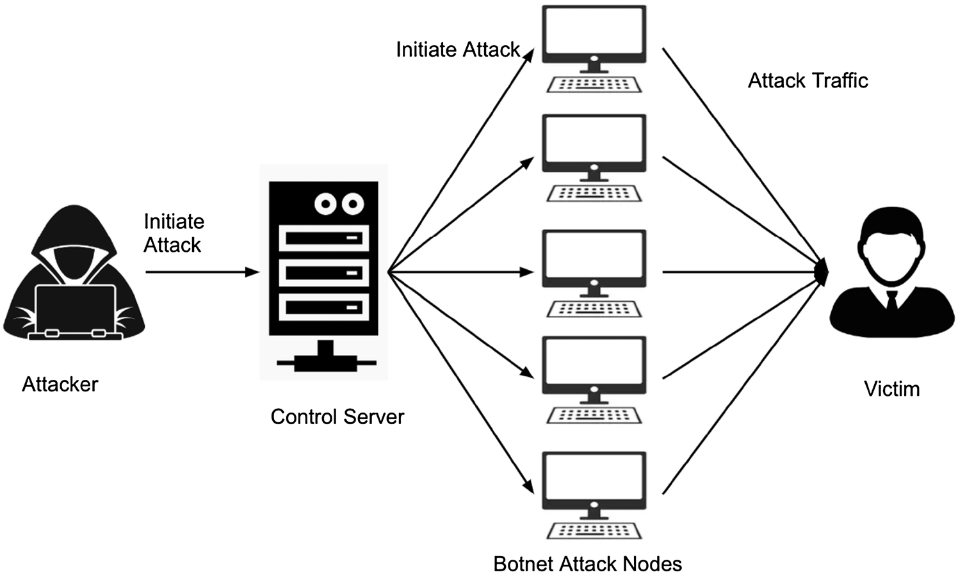
Distributed Denial of Service (DDoS) attacks are the important threats to the Internet because number of clients are sending the request to single server, so server are not able to provide the proper services to the clients due to excessive resource consumption [1–3]. The terms 'Bot' and 'Botnet' originated from Internet Relay Chat (IRC) which uses a central structure on a single settled server port [4]. They aim towards spreading infection through compromised systems [5]. Botnets are responsible for several security issues like executing the DDoS attacks, spreading spam, organizing snap bending traps, taking individual client data (credit card information, government powerlessness information) and abusing extraordinary computational assets [6].

A Botnet is accumulation of PCs related with the Internet which have been wrangled and controlled remotely by an intruder, usually called as a Bot-master, for destructive programming [7]. Moreover, it may reasonably uncover messages being passed from the server to singular customers [8], b,), which poses further security risks. IRC-built bots are feeble in the sense that the whole Botnet can be hazardous. It may end up closing down the entire IRC server [9].

Here in Fig. 1, as illustrated, attacker communicates with the Control Server to set up command and control the system. A Control Server is a powerful server with lots of resources, which may be in the form of bandwidth, memory, and processing power. In addition to receiving commands from an attacker, the handlers, also known as Agents are responsible for tracking Botnets. They send commands related to configuration and updates to the Botnets. The owner of the compromised computer system is not aware of the malware installed on their computer or if they are part of the Botnet. The attackers use the agents as a desk Jumps to launch attacks against the targets (victim) [10].

Thus, it is important to detect Botnet DDoS attacks so as to prevent systems and resources from being vandalized. Machine Learning methods when exposed to data are capable of adapting independently and learning from earlier computations in order to interpret the available data for identifying hidden patterns. They have been known to classify network traffic [11], perform software metrics prediction [12], predicting software change proneness [13], Botnet detection [14], DDoS detection [15] etc. We will use this ability of machine learning in order to evaluate performance of Botnet DDoS attack detection. There are several machine learning methods to do the same. With this research, we aim to perform an empirical study of machine learning methods for Botnet DDoS attack detection. We have taken into account machine algorithms like Support Vector Machine (SVM), Artificial Neural Network (ANN), Naïve Bayes (NB), Decision Tree (DT), and Unsupervised Learning (USML) for the study. For validating the performance of these algorithms, two datasets, UNBS-NB-15 and KDD99 have been investigated. The results have been indicated by means of False Alarm Rate (FAR) and Accuracy for both the datasets considering the machine learning algorithms. Based on the results, conclusions have been made. This is significant in computer security and other related fields. The rest of the paper is organized as follows: The background section incorporates taxonomy of bots as well as the literature survey as section II. Section III highlights materials and methods which includes datasets and methodology. Section IV presents results and discussions while section V presents conclusions.

Fig. 1 DDoS Attack



2 Background

2.1 Taxonomy of bots

In this section, we take into account several bots and provide description based on their features (Different Types of bots [16]). Table 1 depicts some different typical types of Bots and their description.

2.2 Literature review

This section highlights the research work related to Botnets that has been carried out in the past few years. We do not limit our literature survey to only DDoS attacks in Botnets have also studied several other implications. Based on our research we perform a critical evaluation of the research that has been already carried out.

Boshmaf et al. [26] evaluated how vulnerable online social networks (OSNs) are to intrusion by a Socialbot Framework (SbN). They used Facebook as a delegate OSN, and found that using bots that duplicate legitimate OSN customers is effective in invading Facebook extensively, especially when customers and bots share normal allies. The major drawbacks of this research were that security defenses may not be adept in detecting such bots and that it led to privacy breaches and ill consequences on other socially-informed software systems. Alomari et al. [27] showed the risk of Botnet-develop DDoS in application layer (web servers). The paper takes into account Botnet based DDoS attacks along with various attack models and tools. It classifies Botnets on basis of DDoS attacks. The primary drawback of this research was that the solution proposed was not optimal and there were scalability constraints. Zhao et al. [28] used machine learning classification techniques to classify network traffic behavior to detect Botnets before they perform any malicious activity. Performance evaluation of two classifiers led to successful detection of Botnets with high accuracy. The limitations of this work are that the technique is dependent on availability of malicious data. Further there is a need to upgrade the classifier with the introduction of new threats. Since network flow is being

Table 1 Bots and their description

No.	Bots	Description
1	Agobot/Phatbot /Forbot/XtremBot [17]	This type of malware is dangerous because it exhibits the ability to distinguish between debugging tools like OllyDbg, SoftICE, etc. and Virtual Machines like Virtual PC, VMWare etc. It is capable of sniffing traffic and can also hide its presence. Reverse Engineering this malware is very difficult. Moreover, the Linux version can recognize the Linux distribution concerning the compromised host
2	SDBot/RBot/ UrBot/UrXBot/…[18]	A remarkable malware, it is capable of providing system remote access to the adversary. SDBot is made in astoundingly poor C and passed under the GPL. It is succeeded by other malwares like RBot, RxBot, UrBot, UrXBot, JrBot etc. Although the command set is limited, and the implementation is simple, it is popular among attackers due to its catastrophic after effects
3	DSNX Bots [19]	Due to the presence of a plugin interface, an attacker may extend the features of this bot by appending scanning and spreading features. The default modification does not incorporate any spreaders, although plugins are used to overcome this limitation. The plugins may also contribute in performing DDoS attacks and port scanning
4	Q8 Bots [20]	It is 926 lines C-code written for Linux/Unix Operating systems. Like most of the bots, it is capable of carrying out flooding attack and execution of arbitrary commands. Some versions of this bot support spreaders
5	Kaiten [21]	Kaiten does not incorporate spreaders. Written for Unix/Linux system, this malware takes advantage of weak user authentication. Being a single file, it is easy to fetch and compile. Moreover, it offers remote shell, thus it can use IRC to discover system vulnerabilities and gain privileged access
6	Perl-based bots [22]	These bots contain few hundred lines of code. They are used for Unix-based systems and have a limited set of commands
7	Mirai Ceron et al. [23]	Mirai is a self-propagating worm that replicates itself by finding, attacking and infecting vulnerable to IoT devices. These bots use massive DDoS attacks in order to take down major websites as they are capable of compromising hundreds of thousands of IoT devices. They infect poorly secured Internet devices by using telnet to find which one are still using the factory default usernames and passwords. The harm of Mirai is due to its ability to infect a dozen of thousands of other poorly secured devices and enable them to execute a DDoS attack against a chosen target
8	GameOver Zeus [24]	It is a peer-to-peer Botnet that derives characteristics from ZeuS trojan. It is used by scammers to control and monitor data through the Command and Control (C&C) server. The underlying idea is that the virus establishes a connection to the server once executed and disables several processes running on the system. It may cause hindrance for launching processes and downloads and may also delete files
9	Pushdo [25]	The Pushdo Botnet is used for spamming and is observed during Distributed Denial of Service (DDoS) attacks against Secure Socket Layer (SSL) enabled websites. It may completely compromise the target system which may lead to exposure of confidential information and loss of productivity

taken into consideration, bot may be capable of evading its detection. Garasia et al. [29] proposed a technique to detect Botnets using timestamps and frequent pattern sets promoted by the Apriori algorithm. They use the technique of traffic monitoring, filtering, network traffic separator and malicious activity detector to detect Botnets based on the Apriori algorithm. The technique has its own limitations like slow time, often leads to data congestion. Bilge et al. [30] proposed yet another technique to detect Botnets using NetFlow Analysis based on feature selection and classification. The drawback of this research was that there were several constraints and assumptions. They did not include packet payload, the collected data may be sampled. Thapngam et al. [31] used traffic pattern analysis to identify Distributed Denial of Service traffic from legitimate traffic using Pearson's correlation coefficient. The method proved to be successful on the datasets considered, however, whether it is feasible for detecting traffic in real time or not is still a question.

Feizollah et al. [32] considered a survey of five machine learning classifiers using malware datasets from Android Malware Genome Assignment over a year. The researcher certifies that machine learning classifier; k-nearest neighbor is the most optimum among all other classifiers. The only drawback this research might face would be due to innovation of new malwares in given time, which would make it mandatory to collect samples every time in order to analyze the performance. Research by Zhao et al. [33] on Botnet detection by classifying behavior based on time intervals on network traffic faced the same issue. Since they are using existing datasets, the performance is subject to modification in case a different data set is considered. Khattak et al. [34] described three broad logical classifications of Botnet based on behavioral features, detection and defense, combined together to form a comprehensive framework to provide a solution to Botnet issues. One of the limitations of this proposed model is lack of long term efficacy. Lim et al. [35] suggested that software-defined network (SDN) may be beneficial in blocking DDoS attacks launched by Botnets. This is carried out by a DDoS blocking application that works along with SDN controller. The limitation of the research is that even though the technique successfully blocks DDoS attacks, the server is not protected. Hoque et al. [36] presented layout of DDoS attacks, their causes and logical classification and specific hidden components of various strike pushing gadgets. There are several drawbacks to this research. The specified method is not robust; hence the attack may not be detected in real time. Performance evaluation may not be easy for the proposed method, keeping in mind multiple user parameters and datasets. Sieklik et al. [37] evaluated Trivial File Transfer Protocol (TFTP) DDoS amplification attack and presented the mitigation strategies. Several amplification factors have been taken into consideration for performance evaluation since there is no common definition for Amplification factor. Also, different TFTP implementations could lead to different amplification factors. Moreover, the research does not take into account packet loss or transmission which would affect the evaluation. The proposed equation does not guarantee accuracy in terms of results.

Stevanovic and Pedersen [38] listed some bot detection methods that take into account machine learning in order to identify Botnet network traffic. The study highlights supervised learning methods like SVM (Support Vector Machines), ANN (Artificial Neural Networks), Decision tree classifiers and Bayesian classifier and some unsupervised learning methods like K-means, X-means and Hierarchical clustering for Botnet traffic detection. Some characteristics that are investigated for the same pertain to point of traffic monitoring and detection target. Characteristics like Botnet type, operational phase and communication protocol have also been mentioned. The evaluation is based on precision, recall, True Positive Rate, False Positive Rate, Accuracy, Error Rate etc. There is substantial cost of error involved due to the high cost of false negatives. This could lead to technical and financial damage.

Sahay et al. [39] proposed a DDoS defense framework for autonomic DDoS mitigation. The framework is known as ArOMA, which is capable of network monitoring and anomaly detection and uses SDN to mitigate DDoS attacks. Scalability could be an issue, in a situation where a large number of mitigation requests would need to be handled. Antonakakis et al. [40] analyzed the growth of Mirai Botnet up to 600 k infections. The paper highlighted how Botnets emerged, which devices were affected by the Botnet, the risks involved in the devices and their probable solutions. The drawback of this research is that it is based on Internet-of-Things (IoT) devices and they are prone to several unique security challenges. As IoT continues to expand, several more security challenges may be expected in the future. Wang et al. [41] detected Domain Generation Algorithm - DGA based Botnets using clustering. DGA based Botnets are difficult to detect since they depend on the behavior of DNS traffic. The method proposed is evaluated using DNS data of an educational environment over 2 years. Although the results are accurate, they are specific to the data set. Other works on the extension of Botnet on mobile devices can be referred in [8, 42–45].

Fok et al. [46] suggested a Botnet traffic detection technique based on machine learning. The research relies on multilayer perceptrons and decision trees on network traffic analysis for detecting traffic automatically. The researchers have used recall and false positive rate (FPR) to justify the results. The results specify that use of Decision Trees instead of the existing threshold-based decision maker may be used effectively to increase the recall of the framework, while the FPR is reduced to almost zero.

Homayoun et al. [47] used a Botnet traffic analyser based on deep learning approach for detecting Botnet traffic i.e. the Botnet Traffic Shark. The functioning of the analyser is based on network transactions and does not take into account deep packet inspection technique. Performance evaluation is based on parameters True Positive Rate and False Positive Rate. The study shows that Autoencoders perform better than Convolution Neural Networks due to smaller false positives generated. Deep learning requires large amount of data for progressive learning. The processing power required is also significantly high.

We have reviewed the research work already performed over the last few years. We have also highlighted various limitations faced by the researchers in the past. This would be supportive of the research work that we wish to conduct in this paper. The following Table 2 shows the summarized critical evaluation of the research work presented in the past years.

In the literature survey conducted, we came across several of the research works that have been done related to detection of **Botnet**s, DDoS attacks, network traffic etc. Most of the research conducted is based on machine learning. Based on the previous work, we aim to study the characteristics of classifiers for detection of DDoS attacks using two different datasets. As we have already mentioned the limitations of the previous research work by means of critical evaluation, we list the strengths of the classifiers we have taken into consideration.

Support Vector Machines One of the major issues that we came across during critical evaluation of literature survey was based on optimization and feasibility of the research. Support Vector Machines overcome that issue. Moreover, it also avoids overfitting and is useful for solving complex problems. Thus, the risk of overfitting in SVMs is less. It has good scalability and is also memory efficient.

Decision Trees Compared to many machine learning algorithms, decision trees require less effort for data preparation while re-processing. It does not require normalizing data or scaling. If values are missing, the process of building decision tree is not affected.

Naïve Bayes A remarkable issue of the previous research works revolves around training time being too high. In Naïve Bayes classifier, a lot of training data is not required. Further, it is easy to implement.

Artificial Neural Networks Artificial Neural Networks are known for storing information in the networks rather than any database. They can work with incomplete knowledge (data). They are fault tolerant and are also capable of parallel processing, which leads to using less memory and resources.

Unsupervised Machine Learning One of the most important applications of Unsupervised Machine Learning is anomaly detection and clustering. The idea is to model hidden patterns or underlying structures in the given input data in order to learn about the data. It may prove useful for performance evaluation of DDoS attacks.

3 Materials and methods

3.1 Datasets

In order to validate the performance of algorithms, we use the benchmark UNBS-NB-15 dataset- one of the latest and widely used datasets. Therefore, it gives precise representation of both traditional network traffic and several network attacks made by Botnets (Table 3). The dataset was created by taking advantage of IXIA Perfect Storm tool, which produced a combination of authorized client and attacked traffic [48, 49], which classified into nine groups: Fuzzers, Backdoor, DoS, DoS, Exploits, Shellcode, Worms, Generic, Reconnaissance and Analysis. Besides, the KDD99 (Table 4) dataset is also selected as an alternative source for testing (The CAIDA UCSD Dataset 2008-11-21 [50].

A short description of the attacks in the datasets is given in Table 5

3.2 Methodology

We came across several techniques of attack detection in Botnets during the literature survey. The techniques proposed pose several limitations pertaining to scalability, large data sets, accuracy, complex data, slow results etc. Hence there is a need to consider a much more feasible technique that addresses all these challenges. Machine Learning is one such approach and it incorporates several techniques. We conduct the performance analysis of some of the most typical machine learning methods used in Botnet DDoS attack detection like SVM, NB, ANN, DT and Unsupervised learning (USML) (K-means, X-means etc.) The reason we have considered these machine learning methods is based on the advantages these methods have, and their appropriateness considering the kind of data we are dealing with for performance assessment. Most of the techniques proposed in the past, face the issue of scalability and robustness, thus we address the issue by considering the machine learning methods that are highly scalable. In order to conduct the performance evaluation, we outline a framework as shown in Fig. 2.

In what follows, we describe the components of the framework:

Collection of Traffic Packet analyzing tool 'tcpdump' is used to capture packets being accessed by Network Interface Card (NIC) so as to generate features. And by using the tcpdump tool on the edge routers as shown in Fig. 3, the tcpdump tools can gather data as usual in the attacks targeted at server resources within the intranet. Although we

Table 2 Critical Evaluation

S/No	Authors	Methods	Limitations
1	Boshmaf et. al. [26]	Vulnerability of OSNs to intrusion by a Socialbot Framework (SbN)	Ineffectual security defenses, privacy breaches, ill consequences on other socially-informed software systems
2	Alomari et. al. [27]	Risk of Botnet-based DDoS prevailing in application layer	Scalability constraints, non-optimal solution
3	Zhao et al. [28]	Detection of Botnets using machine learning classification techniques to classify network traffic behavior	Proposed solution is dependent on availability of malicious data. Need to upgrade the classifier with new threats. Bot may be capable of evading its detection
4	Garasia et al. [29]	Technique to detect Botnets using timestamps and frequent pattern (Apriori algorithm)	Slow time and data congestion
5	Bilge et al. [30]	Technique to detect Botnets using NetFlow Analysis based on feature selection and classification	Based on several assumptions. They did not include packet payload for evaluation
	Thapngam et al. [31]	Traffic pattern analysis to detect DDoS traffic	Feasibility of the proposed method in question for real time data
6	Feizollah et. al. [22]	Conducted survey of five machine learning classifiers to find the most optimum classifier	Performance analysis (new malwares) requires updated data samples to be collected
7	Zhao et al. [33]	Botnet detection by classifying behavior based on time intervals on network traffic	Different performance for different data sets, thereby offering no concrete evaluation results or solution
8	Khattak et al. [34]	Classifications of Botnet based on behavioral features, detection and defense (comprehensive framework)	Long term efficacy
9	Lim et al. [35]	Software-defined network (SDN) technique to block DDoS attacks launched by Botnets	Unprotected server
10	Hoque et al. [36]	DDoS attacks, their causes and logical classification and specific hidden components	Lacks robustness, impossible to detect attacks in real time, multiple user parameters and datasets lead to inaccurate performance evaluation
11	Sieklik et al. [37]	TFTP DDoS amplification attack, mitigation strategies	Possible ambiguous results due to absence of a proper definition for Amplification Factor, research does not consider packet loss or transmission, inaccurate results
12	Stevanovic and Pedersen [38]	Bot detection method using machine learning in order to identify Botnet network traffic.	There is substantial cost of error involved due to high cost of false negatives. This could lead to technical and financial damage
14	Sahay et al. [39]	DDoS defense framework for autonomic DDoS mitigation (ArOMA)	Scalability issue
15	Antonakakis et al. [40]	Analysis of Mirai Botnet up to 600 k infections	Completely based on IoT devices which are prone to several unique security challenges
16	Wang et al. [41]	Detected DGA based Botnets using clustering	Results are specific to dataset
17	Fok et al.[46]	Automated Botnet Traffic Detection using multilayer perceptrons and decision trees	The method does not produce the best results
18	Homayoun et al. [47]	Deep learning approach to detect Botnet traffic using Botnet Traffic Shark	Deep learning has its own disadvantages in form of requirement of huge amount of data and large amounts of processing power

Table 3 UNBS-NB 15 dataset

Ranking	Selection of features	Feature description
0.642	Sbytes	Client to server transactions
0.491	Dbytes	Server to client transactions
0.477	Smean	Average size packet transmitted by client
0.464	Sload	Client bits/second
0.454	ct_state_ttl	Count
0.444	Sttl	Client to server (time)
0.439	Dttl	Server to client (time)
0.429	Rate	Rate
0.409	Dur	Recorded time duration
0.406	Dmean	Average packet size transmitted by server

collect data from the mentioned datasets, tcpdump is only a method of capturing packets to support the framework. We use common tools like Bro and Argus to generate features in the UNBS-NB 15 dataset from raw packets. The network sniffing procedure wants to be conceded at the main point of the network to identify the flow transferring through the router. IP address and protocol are used for determining the features. The process consists of discovering the source of network-related incidents, and reducing the required processing time (Fig. 4).

Selection of Genuine Features The methods are classified into filter, wrapper and hybrid. Filter method is a preprocessing step wherein features are selected on the basis of scores. These scores are the outcomes of specific statistical tests. Information Gain (IG) and Chi square (x^2) are some of the filtering methods.

3.2.1 Machine learning procedures

Support Vector Machine (*SVM*) Assume a set of training samples, each of the elements in the set marked as fitting to one of two classes, an SVM algorithm creates a model to guess whether a new sample falls into one of the two classes [51]. Both non-spoofed and spoofed IP can be detected using this approach. Enhanced SVM is used to detect non-spoofed IP and Hop Count Filtering mechanism to detect spoofed IP. The Enhanced Multiclass SVM is used for detection of the attacks into various classes for a generated dataset and SVM is used for the assessment of EMCSVM. One of the main reasons for considering this technique is the strength of classification if provides and precision of results it guarantees. It also complements IG pertaining to the classification techniques.

Table 4 KDD99 dataset

S/No.	Features	Descriptions
1	Duration	Connection time between source to destination (in Seconds)
2	Protocol_type	Types of protocol that used during the connection
3	Service	Service on the destination
4	Flag	Status flag of the connection
5	Src_bytes	Data (Bytes) send from source to destination
6	Dst_bytes	Data (Bytes) send from destination to source
7	Wrong_fragment	List of wrong fragment from source to destination and vice versa
8	Urgent	Number of urgent packets from source to destination and vice versa
9	Land	1 if the connection from the same host, 0 otherwise

Table 5 Attack and description

No.	Attack	Description
1	*Fuzzers*	Attacker aims at determining security holes in a system by using the combination of attacks
2	*Backdoor*	Altering authentication mechanisms, allowing a user break into and control the infrastructure without being discovered
3	*DoS*	An intervention technique, aims to stop services from the network and computer resources
4	*Exploit*	It makes use of bugs in the code, its target server and clients
5	*Shell Code*	In this attack, attackers are interred a code in the running system with the help of the internet
6	*Worm*	It makes a copy of itself, thus distributing itself into a single or multiple hosts
7	*Generic*	An attacker tries to crash a system with the help of hash function
8	*Reconnaissance*	A data gathering occurs before real attack
9	*Analysis*	A combination of various attack techniques to attacks targeting

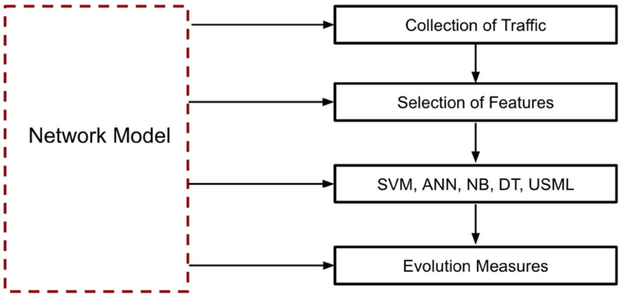


Fig. 2 Proposed framework for analyzing Botnet DDoS Attack

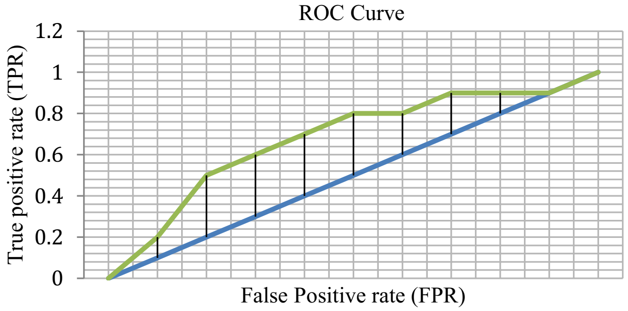


Fig. 4 ROC curve between TPR and FPR

Artificial Neural Network (ANN) it is based on human neurons, a hybrid neural network consists of a self-organizing map (SOM) and radial basis functions to identify and classify DDoS attacks. They can not only effectuate both linear and non-linear data, but also ensure scalability [52]. They can commit to high processing and storage of data and have the ability to self-repair. They are also relied on for their fault tolerance capabilities.

Naïve Bayes (*NB*) It classifies a record R_1 which is often the collection of features into a specific class C_2, if and only if the probability of that record belongs to that specific class, this means, $P\left(\frac{C2}{R1}\right) > P\left(\frac{Cn}{R1}\right)$, with C_n being any class rather than C_2. This classifier requires quite less training data and is highly extensible [53].

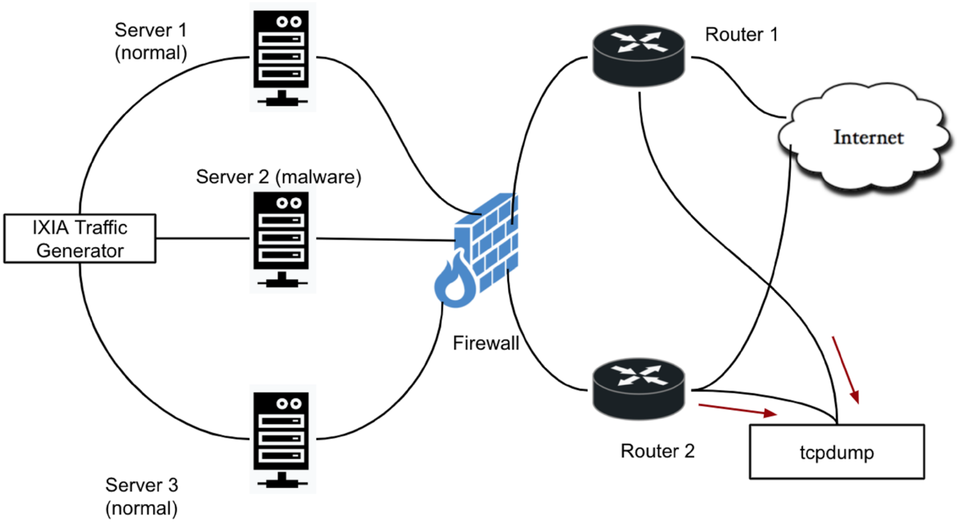
Decision Tree (DT) In order to determine the class chosen for the record, it produces a tree like structure. They contribute to feature selection due to their tree-like structure consisting of nodes. They are expandable and require less effort from users [54].

Unsupervised learning (*USML*) Unsupervised learning is used to discover social affairs of relative cases inside the data. K-means and X-means are some of the most popular unsupervised learning approaches used for Botnet detection. It can handle larger and complex models [55].

4 Results and discussion

In our research work, we have relied on five machine learning algorithms, namely Support Vector Machines, Artificial Neural Networks, Naïve Bayes, Decision Tree and Unsupervised Learning in order to draw comparison by using datasets UNBS-NB 15 and KDD99.In order to assess the results, we make use of the confusion matrix which is used for comparing the performance of the algorithms. It usually consists of a table describing the possible results of a classification. In our case, the results are in the form of '1', which denotes that there was an attack detected or '0' which denotes normal network traffic, against the actual values of the class feature already existing in the evaluation (testing) dataset. The confusion matrix may show four conditions:

Fig. 3 Experimental model using tcpdump tool to collect the UNSW-NB 15 dataset



True Positive (TP) The classifier has accurately determined the class feature in which attack was detected.

True Negative (TN) The value of the class feature is negative, i.e., normal traffic.

False Positive (FP) The classifier incorrectly identifies a normal traffic as an attack.

False Negative (FN) The classifier incorrectly classifies an attack record as normal traffic.

With the help of these conditions, we can create seven metrics called Accuracy, False Alarm Rate (FAR), Sensitivity, Specificity, False positive rate (FPR), AUC, and Matthews correlation coefficient (MCC), it can assess the Classifiers. These two metrics are calculated as follows: Accuracy may be represented by the probability that a record is accurately identified, which can either be an attack or normal traffic. The calculation of overall accuracy is shown as:

$$Accuracy = \frac{TN + TP}{TP + TN + FN + FN} \tag{1}$$

FAR represented the probability in which a record is incorrectly classified.

$$FAR = \frac{FP + FN}{FP + FN + TP + TN} \tag{2}$$

Sensitivity, evaluate the effect of uncertainty in each uncertain computer input on a particular model output.

$$Sensitivity = \frac{TP}{TP + FN} = Truepositiverate(TPR) \tag{3}$$

Specificity, represent the probability of test attacks without giving false positive results.

$$Specificity = \frac{TN}{TN + FP} \tag{4}$$

Table 6 Accuracy classification by AUC

S/No.	AUC Range	Classifications
1	0.90 < AUC < 1.00	Very good
2	0.80 < AUC < 0.90	Good
3	0.70 < AUC < 0.80	Poor
4	0.60 < AUC < 0.70	Very poor

False positive rate (FPR), can be measured as:

$$FPR = \frac{FP}{FP + TN} = 1 - Specificity \tag{5}$$

All the possible combination of TPR and FPR compose the ROC space, the area under the ROC curve AUC (Table 6) measure the accuracy. The AUC of ROC curve can be measured by the following equation:

$$AUC = \int_0^1 ROC(t)dt \tag{6}$$

where t=1-Specificty, and ROC(t) is sensitivity.

ROC curve shows the description between the TPR and FPR, by both the coordinates (0, 1), testing diagonal point with 50% TPR and 50% FPR between the coordinate (0, 0) and (1, 0).

Matthews correlation coefficient (MCC), is used in machine learning for measuring quality of two binary classifications and may be defined as

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{7}$$

We conduct the experiment, so as to present the results graphically. It is a Classifier v/s Accuracy, False Alarm Rate (FAR), Sensitivity, Specificity, False positive rate (FPR), AUC, and Matthews correlation coefficient (MCC). The classifiers are SVM, DT, NB, ANN and USML, whereas the Accuracy and FAR are based on percentage values (1 to 100) and for validation all the methods we used 10-Fold cross validation [56] for both datasets UNBS-NB15 and KDD99. The table depicting the performance evolution for UNBS-NB15 dataset is presented in (Table 7).

As is evident from the performance evaluation results, accuracy for Unsupervised Machine learning is the highest followed by Decision trees and Support Vector Machines. Naïve Bayes and Artificial Neural Networks are not very accurate. Correspondingly their FAR values increase. While USML has the lowest FAR value, ANN has the highest. Based on the experimental analysis, we observe that USML (unsupervised learning) is the best at differentiating with Botnet and normal network traffic with an accuracy of 94.78%. These algorithms make uses of Information Gain for splitting the data by using classification

Table 7 Performance evolution for UNBS-NB 15 dataset

Classifier	Accuracy (%)	FAR (%)	Sensitivity (%)	Specificity (%)	FPR (%)	AUC (%)	MCC (%)
SVM	84.32	15.68	99.08	0.92	92.8	92.45	7.55
DT	94.43	5.57	94.52	5.48	94.52	96.52	3.48
NB	71.63	28.37	93.45	6.55	93.45	84.68	15.32
ANN	63.97	36.03	96.84	3.16	96.84	89.65	10.35
USML	94.78	5.22	89.78	10.22	89.78	96.57	3.43

Table 8 Performance evolution for KDD99 dataset

Classifier	Accuracy (%)	FAR (%)	Sensitivity (%)	Specificity (%)	FPR (%)	AUC (%)	MCC (%)
SVM	91.55	8.45	90.13	9.87	90.13	89.54	10.46
DT	93.3	6.7	93.14	6.86	93.14	94.52	5.48
NB	96.74	3.26	98.21	1.71	98.29	89.58	10.42
ANN	97.44	2.56	84.89	15.11	84.89	85.54	14.46
USML	98.08	1.92	91.88	8.12	91.88	98.52	1.48

feature, accuracy USML (94.78%), lowest FAR (5.22%), sensitivity (89.78%), Specificity (10.22%), FPR (89.78%), AUC (96.57%) and MCC (3.43%). Table 8 indicated the Accuracy, False Alarm Rate (FAR), Sensitivity, Specificity, False positive rate (FPR), AUC, and Matthews correlation coefficient (MCC) for USML.

We perform a similar experimental analysis on KDD99 dataset [57]. While the accuracy of USML (98.08%), Artificial Neural Networks, Naïve Bayes, Decision Trees and Support Vector Machines are less accurate respectively with their accuracy percentage being 97.44%, 96.74%, 93.3% and 91.55%. As far as FAR is considered, the value for USML is the least (1.92%), and Sensitivity (91.88%), Specificity (8.12%), False positive rate (91.88%), AUC (98.52%) and MCC (1.48%) depicting that it is the best classifier even for this dataset. Table 8 indicates the accuracy and FAR for KDD99.

5 Conclusion

In this paper, we analyzed machine learning algorithms for Botnet DDoS attack detection. The tested algorithms are SVM, ANN, NB, DT, and USML (K-means, X-means, etc.). The evaluation was done on the UNBS-NB 15 and KDD99 datasets, which are well-known publicity for Botnet DDoS attack detection. It has been shown that USML (unsupervised learning) is the best at differentiating between Botnet and normal network traffic in term of Accuracy, False Alarm Rate (FAR), Sensitivity, Specificity, False positive rate (FPR), AUC and MCC. This validation is significant in computer security and other related fields.

In the future, several other data sets may be taken into account to verify the credibility of the machine learning methods. This paper considered the Distributed Denial of Service (DDoS) attacks only. Thus, several other attacks may be studied under the same approach. New machine learning algorithms base on neutrosophic theory [58, 59, 60] may be introduced for specific types of attacks.

Compliance with ethical standards

Conflict of interest The authors declare that they do not have any conflict of interests.

Human and animal rights This research does not involve any human or animal participation. All authors have checked and agreed the submission.

References

1. Al-Jarrah OY, Alhussein O, Yoo PD, Muhaidat S, Taha K, Kim K (2016) Data randomization and cluster-based partitioning for Botnet intrusion detection. IEEE Trans Cybern 46(8):1796–1806
2. Bhushan K, Gupta BB (2018) Distributed denial of service (DDoS) attack mitigation in software defined network (SDN)-based cloud computing environment. J Ambient Intell Humaniz Comput. https://doi.org/10.1007/s12652-018-0800-9
3. Tom Ball (2018) Malicious Botnets responsible for 40% of global login attempts. https://www.cbronline.com/news/malicious-Botnets-login
4. Nadji Y, Antonakakis M, Perdisci R, Dagon D, Lee W (2013) Beheading hydras: performing effective Botnet takedowns. In: Proceedings of the 2013 ACM SIGSAC conference on computer & communications security, pp 121–132
5. Cao N, Li G, Zhu P, Sun Q, Wang Y, Li J, Zhao Y (2018) Handling the adversarial attacks. J Ambient Intell Humaniz Comput 1–15
6. Singh K, Guntuku SC, Thakur A, Hota C (2014) Big data analytics framework for peer-to-peer Botnet detection using random forests. Inf Sci 278:488–497
7. Karim A, Salleh RB, Shiraz M, Shah SAA, Awan I, Anuar NB (2014) Botnet detection techniques: review, future trends, and issues. J Zhejiang Univ Sci C 15(11):943–983
8. Pillutla H, Arjunan A (2018) Fuzzy self organizing maps-based DDoS mitigation mechanism for software defined networking in cloud computing. J Ambient Intell Humaniz Comput. https://doi.org/10.1007/s12652-018-0754-y
9. Beitollahi H, Deconinck G (2014) Connection score: a statistical technique to resist application-layer ddos attacks. J Ambient Intell Humaniz Comput 5(3):425–442
10. Rodríguez-Gómez RA, Maciá-Fernández G, García-Teodoro P (2013) Survey and taxonomy of Botnet research through life-cycle. ACM Comput Surv (CSUR) 45(4):45
11. Reza M, Sobouti M, Raouf S, Javidan R (2016) Network traffic classification using machine learning techniques over software defined networks. Int J Adv Comput Sci Appl 8(7):220–225

12. Jha S, Kumar R, Son L, Abdel-Basset M, Priyadarshini I, Sharma R, Long H (2019) Deep learning approach for software maintainability metrics prediction. IEEE Access 7:61840–61855
13. Pritam N, Khari M, Son L, Kumar R, Jha S, Priyadarshini I, Abdel-Basset M, Long H (2019) Assessment of code smell for predicting class change proneness using machine learning. IEEE Access 7:37414–37425
14. Hoang X, Nguyen Q (2018) Botnet detection based on machine learning techniques using DNS query data. Future Internet MDPI 10(5):43
15. Zekri M, Kafhali S, Aboutabit N, Saadi Y (2017) DDoS attack detection using machine learning techniques in cloud computing environments. In: 3rd international conference of cloud computing technologies and applications (CloudTech), pp 1–7. https://doi.org/10.1109/cloudtech.2017.8284731
16. Different types of bots. Retrieved from https://www.honeynet.org/book/export/html/53
17. Sarwar S, Zahoory A, Zahra A, Tariq S, Ahmed A (2014) BOTNET—threats and countermeasures. Int J Sci Res Develop 1(12):2682–2683
18. Gu G, Yegneswaran V, Porras P, Stoll J, Lee W (2009) Active Botnet probing to identify obscure command and control channels. In: Annual computer security applications conference, IEEE, pp 1–13
19. Erbacher R, Cutler A, Banerjee P, Marshall J (2008) A multi-layered approach to Botnet detection. In: 2007, proceedings of the 2008 international conference on security & management, SAM, 30:1–308
20. Wolff R, Hobert S, Schumann M (2019) How may i help you?—state of the art and open research questions for chatbots at the digital workplace. In: Hawaii international conference on system sciences, pp 95–104
21. Lu W, Tavallaee M, Ghorbani A (2009) Automatic discovery of Botnet communities on large-scale communication networks. In: Proceedings of the 4th international symposium on information, computer, and communications security, pp 1–10
22. Gupta S, Borkar D, Mello C, Patil S (2015) An E-commerce website based chatbot. Int J Comput Sci Inf Technol 6(2):1483–1485
23. Ceron J, Jessen K, Hoepers C, Granville L, Margi C (2019) Improving IoT Botnet investigation using an adaptive network layer. Sens MDPI 19(3):727
24. Andriesse D, Rossow C, Stone-Gross B, Plohmann D, Bos H (2013) Highly resilient peer-to-peer Botnets are here: an analysis of Gameover Zeus. In: 2013 8th international conference on malicious and unwanted software [proceedings]: "The Americas", MALWARE 2013. [6703693], ACM, IEEE Computer Society, Fajardo, pp 116–123
25. John J, Moshchuk A, Gribble S, Krishnamurthy A (2009) Studying spamming Botnets using Botlab. In: Proceedings of the 6th USENIX symposium on Networked systems design and implementation, pp 291–306
26. Boshmaf Y, Muslukhov I, Beznosov K, Ripeanu M (2013) Design and analysis of a social Botnet. Comput Netw 57(2):556–578
27. Alomari E, Manickam S, Gupta BB, Karuppayah S, Alfaris R (2012) Botnet-based distributed denial of service (DDoS) attacks on web servers: classification and art. Preprint arXiv:1208.0403
28. Zhao D, Traore I, Ghorbani A, Sayed B, Saad S, Lu W (2012) Peer to peer Botnet detection based on flow intervals. Inf Secur Priv Res 87–102
29. Garasia SS, Rana DP, Mehta RG (2012) HTTP Botnet detection using frequent patternset mining. Proc Int J Eng Sci Adv Technol 2:619–624
30. Bilge L, Balzarotti D, Robertson W, Kirda E, Kruegel C (2012) Disclosure: detecting Botnet command and control servers through large-scale net flow analysis. In: Proceedings of the 28th annual computer security applications conference, ACM, pp 129–138
31. Thapngam T, Yu S, Zhou W, Makki S (2012) Distributed Denial of service (DDoS) detection by traffic pattern analysis. In: Peer-to-Peer networking and applications December 2014, Springer, Vol 7, Issue 4, pp 346–358
32. Feizollah A, Anuar NB, Salleh R, Amalina F, Shamshirband S (2013) A study of machine learning classifiers for anomaly-based mobile Botnet detection. Malaysian J Comput Sci 26(4):251–265
33. Zhao D, Traore I, Sayed B, Lu W, Saad S, & Ghorbani A, Garant D (2013) Botnet detection based on traffic behavior analysis and flow intervals. Comput Secur 39:2–16. https://doi.org/10.1016/j.cose.2013.04.007
34. Khattak S, Ramay NR, Khan KR, Syed AA, Khayam SA (2014) A taxonomy of Botnet behavior, detection, and defense. IEEE Commun Surv Tutor 16(2):898–924
35. Lim S, Ha J, Kim H, Kim Y, Yang S (2014) A SDN-oriented DDoS blocking scheme for Botnet-based attacks. In: 2014 6th international conference on ubiquitous and future networks (ICUFN), IEEE, pp 63–68
36. Hoque N, Bhattacharyya DK, Kalita JK (2015) Botnet in DDoS attacks: trends and challenges. IEEE Commun Surv Tutor 17(4):2242–2270
37. Sieklik B, Macfarlane R, Buchanan WJ (2016) Evaluation of TFTP DDoS amplification attack. Comput Secur 57:67–92
38. Stevanovic M, Pedersen JM (2016) On the use of machine learning for identifying Botnet network traffic. J Cyber Secur Mob 4(2):1–32
39. Sahay R, Blanc G, Zhang Z, Debar H (2017) ArOMA: an SDN based autonomic DDoS mitigation framework. Comput Secur 70:1–18. https://doi.org/10.1016/j.cose.2017.07.008.
40. Antonakakis M, April T, Bailey M, Bernhard M, Bursztein E, Cochran J, Kumar D (2017) Understanding the miraiBotnet. In: USENIX security symposium
41. Wang TS, Lin HT, Cheng WT and Chen CY (2017) DBod: Clustering and detecting DGA-based botnets using DNS traffic analysis. Comput Secur 64:1–15
42. Ali ST, Mc Corry P, Lee PHJ, Hao F (2017) Zombie Coin 2.0: managing next-generation Botnets using Bitcoin. Int J Inf Secur 1–12
43. Anagnostopoulos M, Kambourakis G, Gritzalis S (2016) New facets of mobile Botnet: architecture and evaluation. Int J Inf Secur 15(5):455–473
44. Kirubavathi G, Anitha R (2018) Structural analysis and detection of android Botnets using machine learning techniques. Int J Inf Secur 17(2):153–167
45. Pillutla H, Arjunan A (2018) Fuzzy self organizing maps-based DDoS mitigation mechanism for software defined networking in cloud computing. J Ambient Intell Humaniz Comput 1–13
46. Fok K, Zheng L, Watt K, Su L, Thing V (2018) Automated Botnet traffic detection via machine learning. In: Conference: TENCON 2018
47. Homayoun S, Ahmadzadeh M, Hashemi S, Dehghantanha A, Khayami R (2018) BoTShark: a deep learning approach for Botnet traffic detection. In: Dehghantanha A, Conti M, Dargahi T (eds) Cyber threat intelligence advances in information security, vol 70. Springer, Cham
48. Koroniotis N (2017) Towards developing network forensic mechanism for Botnet activities in the IoT based on machine learning techniques. Preprint arXiv:1711.02825
49. Nour M, Slay J (2015) UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In: Military communications and information systems conference (MilCIS), IEEE
50. The CAIDA UCSD Dataset 2008-11-21 (2008) https://data.caida.org/datasets/security/telescope-3days-conficker/

51. Evgeniou T, Pontil M (2000) Support vector machines: theory and applications. In: 2000, Machine learning and its applications, advanced Lectures, pp 249–257
52. Shiruru K (2016) An introduction to artificial neural network. Int J Adv Res Innov Ideas Edu 1(5):27–30
53. Taheri S, Mammadov M (2013) Learning the naive Bayes classifier with optimization models. Int J Appl Math Comput Sci 23(4):787–795
54. Rokach L, Maimon O (2004) Decision Trees. The data mining and knowledge discovery handbook, In book, pp 165–192
55. Khanum MA, Mahboob T, Imtiaz W, Ghafoor HA, Sehar R (2015) A survey on unsupervised machine learning algorithms for automation, classification and maintenance. Int J Comput Appl 119(13):34–39
56. Rodríguez J, Pérez A, Lozano JA (2010) Sensitivity analysis of k-fold cross validation in prediction error estimation. IEEE Trans Pattern Anal Mach Intell 32:569–575
57. Nour M, Slay J (2016) The evaluation of network anomaly detection systems: statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set. Inf Secur J A Glob Perspect 25(13):18–31
58. Son NTK, Dong NP, Son LH, Long HV (2019) Towards granular calculus of single-valued neutrosophic functions under granular computing. Multimed Tools Appl. https://doi.org/10.1007/s11042-019-7388-8
59. Son NTK, Dong NP, Long HV, Son LH, Khastan A (2019) Linear quadratic regulator problem governed by granular neutrosophic fractional differential equations. ISA Trans. https://doi.org/10.1016/j.isatra.2019.08.006
60. Khan MMT, Singh K, Son LH, Abdel-Basset M, Long HV, Singh SP (2019) A novel and comprehensive trust estimation clustering based approach for large scale wireless sensor networks. IEEE Access 7:58221–58240

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.