Cluster Computing (2019) 22:S423–S434
https://doi.org/10.1007/s10586-018-2181-4

Detection of DoS attacks in cloud networks using intelligent rule based classification system

Rakesh Rajendran[1] · S. V. N. Santhosh Kumar[2] · Yogesh Palanichamy[1] · Kannan Arputharaj[1]

Received: 15 December 2017 / Revised: 1 February 2018 / Accepted: 14 February 2018 / Published online: 23 February 2018
© Springer Science+Business Media, LLC, part of Springer Nature 2018

Abstract
Cloud Network has emerged as one of the most adopted technologies both among the end-users and the developers. Despite cloud networks being popular, security in cloud remains a pivotal research challenge and a topic that is much discussed about. Denial of service (DoS) attack is carried out in cloud by one or more perpetrators using multiple compromised nodes to flood a specific target and thereby resulting in unavailability of services. Classification methods can be used effectively to identify attack signature or recurring patterns of such DoS attacks. Therefore, classification using machine learning techniques have been used in this work for feature selection and classification in order to identify the DoS attacks. For this purpose, a new rule based approach for detecting the DoS attacks which uses a domain expert's knowledge has been proposed in this paper. Moreover, two new algorithms namely Feature Selection Algorithm using Scoring and Ranking and Rule based Classification Algorithm for detecting DoS Attacks are proposed in this paper in which the final classification is carried out by applying the rules from the rule base and is validated using a domain-expert. We have evaluated the proposed system on an experimental set-up on cloud and used real time DoS tools and observed that the proposed method achieved better DoS attack detection accuracy than the existing classification algorithms used for security.

Keywords Cloud network · DoS attacks · Classification · Rule base · Domain expert · Security

1 Introduction

Cloud networks have paved a suitable way for users to access applications, services and resources over the Internet. An end user with the help of an internet connection and a computer can access the applications, request necessary services and resources anywhere through cloud networks. Such a model has made organisations to shift their focus on providing a pay as you go and on demand business from their usual day-to-day running of information technology (IT) services. Such a business model in recent times is deemed beneficial and is popular. The cloud network model can be briefly categorised into Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS) and Infrastructure-as-a-Service (IaaS) models. Moreover, these models can be installed in the end user environment as either a public, private or hybrid cloud. Google and Amazon and other major software service providers have joined the cloud networks bandwagon and are extremely popular after successfully setting up their cloud services [1].

With such advances in cloud networks, the probability of threats is on the rise. Reports and news suggest malign activities such as data loss and hacking in cloud network are increasing day by day. Availability of the cloud services and user privacy in the cloud network is of prime importance when the cloud service providers such as Amazon and Google setup their services. Absence or

✉ S. V. N. Santhosh Kumar
santhoshkumar@saveetha.ac.in;
santoshkumar.kumar34@gmail.com

Rakesh Rajendran
rakesh3929journal@gmail.com

Yogesh Palanichamy
yogesh@annauniv.edu

Kannan Arputharaj
kannan@annauniv.edu

[1] Department of Information Science and Technology, College of Engineering Guindy, Anna University, Chennai 600025, India

[2] Department of Computer Science and Engineering, Saveetha Engineering College, Thandalam, Chennai 602105, India

unavailability of cloud services can be attributed to many reasons. But the major reason is either due to the cloud network service component failing or attacks such as Denial of Service (DoS) that are aimed towards the cloud network servers. Data loss or privacy alone is not the reason for us to effectively counter such DoS attacks. A DoS attack is an organised network attack which is carried out with an aim of disrupting an organisation's service operations by effectively denying their access to their end users. It renders the network services of an organisation inaccessible to its users by denying access to the network service. Due to technological advances, even the tools used to carry out such attacks are also updated and hence it is necessary for those defending such attacks to stay up-to-date with high level defence techniques and to know about the recent models used to capture such attacks. A DoS attack is orchestrated through one or more weak systems usually compromised and held in control by an attacker to repeatedly send malicious requests aimed at a target thereby exhausting the target's resources. Such an attack if executed successfully leads to the unavailability of cloud resources. Based upon the report in [2], it is observed that cloud network platform is one of the most attacked platforms used in the recent years.

Data mining techniques can be effectively used in detection of such attacks in cloud. Classification techniques have been applied in identifying such malign activities in various fields such as Web Security, Intrusion Detection Systems, etc. [3]. Common data mining based intrusion detection methods classify the network traffic as either traffic permissible or malign. The detection methods follow a set of signature patterns, anomaly patterns or a combination of both (hybrid). Signature pattern based intrusion detection methods often use a set of derived malicious network traffic rules or patterns. These are then stored in a knowledge base which is referred and compared against real-time network traffic thereby classifying the traffic as permissible or malign. One pre-requisite for such a method is, the knowledge base must be up-to-date to counter malign packets. High false negatives occur if unknown attack signatures are fed as input to a signature pattern based classification method. This is a major drawback of the signature based method for intrusion detection. Classifiers learn from a set of pre-labelled data use-cases in-order to classify a given test instance into one of the pre-labelled class. Classifiers have two phases namely, the training phase and the testing phase. In training phase, the classifier learns from the given pre-labelled data and during the testing phase, the classifier aims to classify a given test instance into one of the classes. The common output class when it comes to security being legitimate or malign.

In this paper, a new security system called Intelligent Rule based Classification System (IRCS) for detecting DoS attacks in cloud network is proposed. The proposed model uses the knowledge of known patterns present in the knowledge base for making an initial decision. This is validated using the rules provided by a domain experts. Moreover, the Domain-expert validates the decisions by his experience in problem solving on intrusions and also by following the rules declared in the knowledge base for effectively finalizing the classification results to decide on the attacks. For this purpose, two new algorithms namely Feature Selection Algorithm using Scoring and Ranking (FSASR) and Rule based Classification Algorithm for detecting DoS Attacks (RCADA) have been proposed in this paper for developing the intelligent rule based classification algorithm in order to enhance the security of communication in cloud networks. Based on the experiments conducted in this work through simulations, it is proved that the proposed model using feature selection and classification algorithms is able to detect 98.5% of the DoS attacks in cloud networks. Moreover, this proposed model is more secured than the existing security models for cloud networks since the proposed model reduces the false positive rate by increasing the classification accuracy using rules present in the rule base and also by validating with a domain expert.

Common forms of DoS attacks include user datagram protocol (UDP) Flood, SYN Flood, Ping of Death, Slowloris, HTTP attack, internet control message protocol (ICMP) Flood etc. This proposed work concentrates on protocol based DoS attacks that targets or uses specific protocols in order to be successful with the attack. It requires careful observation and study of different DoS attack tools deployed in real-time cloud network environment where both legitimate and malign packets flow through the network. To establish basic or advanced network communications between two or more nodes, TCP and UDP protocols are the most used as both the protocols support data to be exchanged. Almost all the web applications require either TCP or UDP protocol along with IP protocol, in order to establish network communication and exchange data. This leads to attackers targeting and manipulating these two protocols mostly. This paper mainly concentrates on DoS attacks carried out through TCP and UDP protocols.

The rest of the paper is organised as follows: Sect. 2 provides a literature survey of the related works. Section 3 describes the proposed system in detail. Section 4 shows the experimental details on this work also depicts the ensuing results with related discussions. Finally, Sect. 5 gives conclusions on this paper and also suggests some future works.

2 Literature survey

There are many works on cloud networks which provides schemes for performance improvement and security in cloud [2, 4–6]. Moreover, the significant increase in the number of users as well as organizations moving to the cloud network makes it difficult to detect DoS attacks. In this section, we have surveyed the work of authors who had previously used data mining approach to counter DoS attacks. Intrusion detection systems [7, 8] are the major techniques used in a cloud network environment in-order to protect the privacy of the end users as well as add security to the data that resides in the cloud network. Choi et al. [9] proposed an approach that uses map reduce model in-order to effectively mitigate DoS attacks. The work aimed against application layer level hypertext transfer protocol (HTTP) DoS attacks. Snort is a sniffing tool that sniffs network traffic in real time in-order to detect suspicious activity. Their proposed work performed better than Snort tool, resulted in shorter processing time and also identified new attack patterns. Pradeepthi et al. [10] proposed a rule based classifier for detecting DoS attacks on cloud networks by implementing a cloud setup and carrying out tests with DoS attack tools.

Many authors used soft computing techniques for solving problems related to networks which can be applied to cloud networks for providing security and optimal communication [11–13]. Gupta et al. [14] proposed a prevention and intrusion detection system based on rules. It uses both Bayesian approach and data analysis to detect malign traffic with the help of unsupervised learning algorithm thereby safeguarding the cloud network against attacks such as Transmission Control Protocol (TCP) SYN flooding. Khorshed et al. [1] built a cloud network environment where typical DoS attack scenario wasrecreated. The incoming traffic was analysed using a Support Vector Machine (SVM) classifier and it reported malign patterns. Santhosh Kumar et al. [15] proposed a hop by hop authentication mechanism which proved effective against DoS attacks in wireless networks. A study of the numerousclient-side as well as server-side protection mechanisms against malign network patterns and DoS attacks proposed so far by various authors was carried out by Wayne Jansen [16]. Many authors have used the features available in cloud network setup such as statistical modelling, Yu et al. [17] and dynamic resource allocation, Girma et al. [18] to effectively counter network attacks. The level of intrusion in cloud network environment and its severity study was proposed by Arshad et al. [19]. Effective security to communication was provided by many researchers by providing secured and intelligent routing algorithms by applying soft computing techniques [20–24].

Using machine learning methodologies, prediction of the severity due to the attacks was done. Ganapathy et al. [3] studied and surveyed the various classification methodologies used in intrusion detection systems. Sindhu et al. [25] used a decision tree based intrusion detection system that employed a wrapper approach. Chonka et al. [26] proposed a methodology which uses cloud trace back technique in-order to trace the source of the DoS attack. The authors employed a back propagating neural network to detect and filter such DoS attacks. Wu et al. [27] proposed a decision tree based method to detect DoS attacks in network layer. Their methodology used a pattern matching technique to classify traffic that is similar to DoS attack. They studied various attack patterns and created a threshold line as the peak traffic rate under normal circumstances. If network traffic is above the pre-defined threshold line, then the traffic is classified as malign by the authors. In spite of the presence of all these works in the literature, the existing systems are not able to detect the DoS attacks with high accuracy. Hence, new algorithms are proposed in this paper to enhance the detection accuracy by applying rules.

3 Proposed work

In this section, a detailed explanation of the proposed work is given. It explains the system architecture, the proposed algorithms for feature selection and classification namely feature selection algorithm using scoring and ranking (FSASR) and rule based classification algorithm for detecting DoS attacks (RCADA) have been explained.

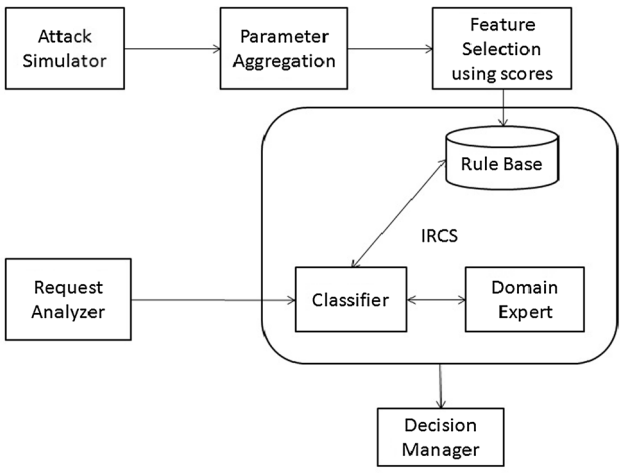

Fig. 1 Architecture of intelligent rule based classification system (IRCS)

3.1 System architecture

The architecture of the intelligent rule based classifier is shown in Fig. 1.

In this proposed architecture, the attack simulator module uses different tools like Low Orbit Ion Cannon (LOIC), XOIC, Pyloris, Ping Flood and SynGUI. The experimental setup is discussed in detail in Sect. 4. A real cloud setup was established and performance metrics with respect to processor usage, memory usage and network bandwidth usage were monitored over a period of time. The values of these metrics differ when the cloud network setup comes under attack due to the traffic generated by the tools as discussed above. The metric values were observed and recorded using TCP dump, process explorer and IPTraf tools. The observed metrics (features) were analysed for their importance and scores were assigned to each feature, as to find the top metrics that impacts the cloud network. The metrics with higher scores were used to form rules for the rule base. When a request arrives, the request is sent to IRCS, where the traffic is analysed and crosschecked by comparing with the stored rules in the rule base to check for the presence of attack patterns. It consults the domain expert during classification. If the request matches any attack pattern, the request is neglected and is not serviced by the decision manager. If IRCS detects any presence of a signature pattern, the process is terminated thereby maintaining the cloud network resources for servicing genuine users. Thus the proposed system effectively manages to thwart any malicious network activity.

3.2 Feature selection algorithm using scoring and ranking

In general, feature selection is carried out to analyse the numerous features available and to identify top features that affects the whole feature set largely with minor or negligible effect with respect to the classification output. It also helps in reducing the size of the feature set, removing redundant and noisy features that might in turn reduce the performance of any classifier. As discussed, performance metrics which affect processor usage, memory usage and network bandwidth usage were observed. The features that were observed during the parameter aggregation phase are given in Table 1.

We formalize the feature selection problem as follows: given is a feature set F, consisting of i features where $F = \{f_1, f_2, f_3,\ldots,f_n\}$. Here each feature is a vector of n values which is given by, $f_i = (v_{i1}, v_{i2},.., v_{in})$. Each feature has a value associated with it, $f_i = V_i$. Certain features among the feature set F, might add noise to the classification results rather than helping to achieve better classification. In order to find top features that give the best classification accuracy, a ranking algorithm can be used. In this kind of problem, if we typically apply a ranking algorithm R such as R (F, P) where P denotes priority, the algorithm gives us a list of features $F = [f_1, f_2, \ldots,f_n]$, ordered by their decreasing priority as output. The factor used to calculate the priority of each feature, Priority (f_i, i = 1 to n) is different for the kind of ranking algorithm used. The algorithm used to rank the features based upon their scores or importance is given below.

Algorithm for Feature ranking using scores

Input : A feature set, $F = (f_1, f_2, f_3,\ldots, f_n)$
Output :Prioritised feature set, F_p

1. initialise$f_{top} = f_1$;
2. **for** i = 2 to n**do**
   a. f_{top}= calculate (f_i,w_i) ;
3. f_j = next (f_1) ;
4. f_{curr} = calculate (f_j, w_i) ;
5. **if**f_{curr} is better than f_{top}**then**
6. **do**
7. $f_{top} = f_{curr}$;
8. **repeat**
9. **end if**
10. until n^{th} feature f_n is evaluated
11. **return** f_{top}

During the application of some existing feature selection methods such as Information Gain, it was observed that features that are important to an attack case were missed when the features were ranked for their priorities. It was found that at times,the priority of a feature fi, was not given due importance as we would like to. Hence, features were ranked based upon their practical importance by adding weights while evaluating a feature f_i. For e.g., consider the scenario discussed previously in the beginning of the proposed work. The feature Valid_user finds whether the request sent by the user is genuine by checking for his registration information in the cloud network. If the user is not a registered, the request can be immediately flagged as suspicious. Hence in the experiments features such as Valid_user, CPU_Usagewould have been assigned more weight than the rest of the features.

After ranking all the 21 features using the above algorithm, the best *'n'* features had to be selected where *'n'* is fixed by the domain expert. Though most of the features that were observed describe the dataset good enough, only few features were redundant and were not considered while

Table 1 List of observed features with description

Features	Description
Valid_USER	A cloud network user registered for the cloud service
Invalid_USER	A random user who hasn't registered for the cloud service
CPU_Usage	Percentage of CPU usage recorded
CPU_Load5	Average percentage of CPU usage recorded for 5 min
CPU_Load10	Average percentage of CPU usage recorded for 10 min
P_DReads	Disk read in progress
P_DWrites	Disk write in progress
P_Handles	Number of system resources that is opened currently
P_Threads	Number of processor threads that is in execution
P_ Local	Processes executing in local service,
P_Network	Processes executing in network service mode
P_User	Processes executing in user mode
P_System	Processes executing in system mode
Mem_WS	The total memory required by a process in a given time
Mem_PrWS	The total memory used by all programs and OS in a given time
Mem_Used	Amount of memory utilised by the system currently
Mem_Buffd	Amount of memory buffered by the system
Mem_Free	Amount of memory that is not used
TCP_Close	TCP Close
TCP_syn	TCP Syn
TCP_W_time	TCP waiting time
IO_Reads	Number of I/O read operations generated by a process
IO_Writes	Number of I/O write operations generated by a process
IO_Read-bytes	Number of read bytes by I/O operations generated by a process
IO_Write-bytes	Number of write bytes by I/O operations generated by a process
Disk_Util	The activity of hard disk utilisation in percentage
KMem_Paged	Amount of kernel memory that can be written to a page file
KMem_Non-Paged	Amount of kernel memory that is kept in RAM itself

forming rules. We follow such a method because the output obtained using a highly ranked feature set is much more reliable. With a larger number of highly ranked features, the accuracy of the classification algorithm also increases. We have used Fisher's discriminant criterion for selecting top *'n'* features from the prioritised feature set F_p, where $n \leq Fp$. We define the problem as follows: To select *'n'* of *'m'* available features with the main aim of achieving maximum class separation. Assume a particular feature, for example, "CPU_Usage". Under different simulated attack scenarios, we know that the value of the feature varies accordingly as given in Table 2.

After normalization, the feature subset looks like the values given in Table 3.

Thus the within class scatter matrix for a feature can be calculated as shown in Eq. (1).

$$F_w = \sum_{i=1}^{n} P(fi) \tag{1}$$

The mean of a feature set and the total mean for multiple feature-set is calculated as shown in Eq. (2).

$$\mu = \sum_{i=1}^{n} \frac{xi}{n} \text{ and } \mu_g = \sum_{i=1}^{n} P(fi)\mu_i \tag{2}$$

The between class scatter matrix for multi-class feature set can be given as Eq. (3).

$$F_b = \sum_{i=1}^{n} P(fi)(\mu i - \mu_g)(\mu i - \mu_g)^T \tag{3}$$

Thus, the feature covariance with respect to the total mean can be calculated asequation (4).

Table 2 Sample feature subset before normalization

Feature	Attack Scenario_1	Attack Scenario_2	Attack Scenario_3
CPU_Usage (%)	86	77	91

Table 3 Sample feature subset after normalization

Feature	Attack Scenario_1	Attack Scenario_2	Attack Scenario_3
CPU_Usage	0.86	0.77	0.91

$$F_c = F_w + F_b \tag{4}$$

Finally, Fisher's discriminant criterion can be given as shown in Eq. (5).

$$FC = \frac{trace(\mathrm{Fw})}{trace(\mathrm{Fb})} \text{ or trace } \left\{F_w^{-1} F_b\right\} \tag{5}$$

Using Fisher's criterion, features that possess good class wise separability are automatically selected for the rule base. It is useful when different attack instances needs to be classified using lesser number of highly ranked features.

3.3 Rule based classification algorithm for detecting DoS attacks

Using the feature selection algorithm as discussed above, rules were formed based upon the priority of the different attack instances that were observed. The list of rules that were formulated for the rule base is given below. The rules given above are generic if–then rules which describe the condition and the certain outcome of the condition if met as shown in the rule give below:

$$\{if\ rule\ 'r_1'\ is\ condition-1,\ if\ rule\ 'r_2'\ is\ condition-2 \ldots if\ rule\ 'r_n'\ is\ condition-n\ then\ class\ is\ C_1\}$$

where r_1, $r_2 \ldots r_n$ are the observed parameter values and C_1 is the attack class it belongs too. Additionally, the following rules are used in this paper for effective detection of DoS attacks.

Proposed Rule I:
ifIO_Reads > IO_Reads (avg) and
ifIO_Read-bytes > IO_Read-bytes (avg) and
ifTCP_W_time > TCP_W_time (avg) and
ifP_Network > P_Network (avg) and
ifP_System > P_System (avg),
then declare pattern as DoSIM attack

Proposed Rule II:
ifIO_Reads > IO_Reads (avg) and
ifIO_Read-bytes > IO_Read-bytes (avg) and
ifCPU_Usage > CPU_Usage (avg) and
if CPU_Load5 > CPU_Load5 (avg) and
if CPU_Load10 > CPU_Load10 (avg),
then declare pattern as SynGUI attack.

Proposed Rule III:
ifP_Handles > P_Handles (avg) and
ifP_Threads > P_Threads (avg) and
ifTCP_W_time > TCP_W_time (avg) and
ifMem_Used > Mem_Used (avg) and
ifMem_Buffd > Mem_Buffd (avg),
then declare pattern as XOIC attack.

Proposed Rule IV:
ifTCP_W_time > TCP_W_time (avg) and
ifMem_Used > Mem_Used (avg) and
ifMem_Buffd > Mem_Buffd (avg) and
ifMem_Free < Mem_Free (avg) and
ifP_Network > P_Network (avg) and
ifCPU_Usage > CPU_Usage (avg) and
if CPU_Load5 > CPU_Load5 (avg) and
if CPU_Load10 > CPU_Load10 (avg),
then declare pattern as LOIC attack.

Proposed Rule V:
ifTCP_W_time > TCP_W_time (avg) and
ifCPU_Usage > CPU_Usage (avg) and
if CPU_Load5 > CPU_Load5 (avg) and
if CPU_Load10 > CPU_Load10 (avg) and
ifIO_Reads > IO_Reads (avg) and
ifIO_Read-bytes > IO_Read-bytes (avg) and
ifP_Handles > P_Handles (avg) and
ifP_Threads > P_Threads (avg),
then declare pattern as Pyloris attack.

In order to arrive at the average values for the different parameters observed the calculation is done as given by Eq. (6).where 'n' is the number of instances of the selected feature.

$$CPU\_Usage\,(avg) = \left\{\frac{CPU\_Usage1 + CPU\_Usage2 \cdots + CPU\_Usagen}{n}\right\} \tag{6}$$

In general, classification algorithms do not consider a domain expert's knowledge while solving a problem. In the proposed methodology, we have formulated Intelligent Rule based Classification System for better accuracy instead of relying upon generic classification algorithms for decision making. It is well known that a domain expert's knowledge for classification tasks improves the results obtained.The detailed steps of the proposed algorithm

namely the Rule based Classification algorithm for detecting DoS attacks for the proposed system is given below:

U – User; **R** – Request; **Q** – Queue; **OP** – Observed Parameters; **rb** – Rule base; **DM** – Decision Manager
Input :U_iR_j : i = 1 to n , j = 1 to m;
Output : DM = U_iR_j: M or L where M: Malign , L: Legitimate

1. **if** U_i is invalid user **then**
   reject U_i
2. **else if** U_i is valid user **then**
3. **for** i= 1 to n
4. **for** j = 1 to m
5. OP_{ij} = analyse (U_iR_j);
6. k = compare (OP_{ij},rb);
7. **if** k is true **then**
   flag U_iR_j as M
8. **else**
   flagU_iR_j as L

As explained previously, the proposed system considers a scenario where a user who registers with the cloud network setup alone can use the cloud network resources. Hence, each request when received in the cloud network traffic is checked first for its authenticity. Only the traffic originating from a valid user is allowed. Once the request is received, it is analysed with respect to change in the observed parameters. These observed values are then compared along with the rules in the rule base. If the pattern matches with any rule formulated in the rule base, then the request is flagged as malicious or it is flagged as legitimate if it doesn't match with the rules. The decision manager follows it up with the necessary action that needs to be taken. In a simple case, the decision manager either allows the network traffic if it is legitimate or rejects it if it is found to be malicious.

4 Results and discussions

4.1 Cloud network environment

In this section, we illustrate the environment used to accomplish different DoS experiments for study and analysis. The cloud network setup was not deployed and tested in a real environment due to issues such as security and legality. For this proposed work, we however cloned a physical network virtually and it was connected to a real network. An overview of the setup is given diagrammatically in Fig. 2. The setup runs on

a. Dell server (Dell Poweredge M620)
b. Oracle VirtualBox Software
c. Virtual routers connecting different virtual nodes and virtual subnets

The configuration of the experimental cloud network setup is given in Table 6.

Genuine traffic was collected from the real physical network environment. This virtual cloud network environment was setup to conduct the attack experiments using different DoS tools and to study the impact of the attacks on the server (victim). It also minimises the probability of the test network packets flooding out to the regular environment. Third party dataset providers such as DARPA and KDD CUP [28] do not update their datasets with latest DoS attack scenarios. Hence for research purpose we decided to generate the own dataset.Various parameters such as processor usage, memory usage, network bandwidth usage etc. were observed so as to form the initial training dataset. Further analysis on this work was entirely based upon this training dataset. The main objective for us to setup a virtual cloud network environment was to carry out DoS attacks in varying frequency (higher frequency and lower frequency). The virtual environment was setup with the help of Oracle Virtual Box software in order to mimic a physical network.

The setup consisted of clones comprising switches, routers, servers and networks. The virtual nodes comprised of Linux and Windows systems and they were placed on different subnets (virtual). The virtual environment consists of three machines. The first system being a Windows

Fig. 2 Experimental Setup

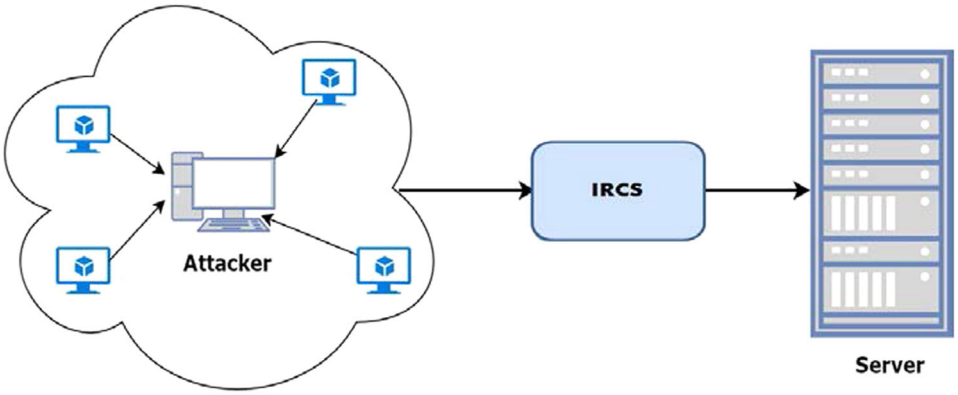

Table 6 System configuration

Item	Configuration
Attacker	Dell PowerEdge M620
OS	Windows Server 2012
CPU	Quad Core Xeon Processor
RAM	64 GB
Server and middle system	Acer Veriton X
OS	Windows 7, Debian
CPU	Intel Core i7,8 Cores
RAM	16 GB
Application (attacker)	Oracle VirtualBox
Network	1GbpsGigabit Ethernet

Server 2012 machine which had VirtualBox software installed in it and was used to create multiple virtual nodes to launch DoS attacks. The second system is a Linux system and it houses the proposed IRCS. The third system is a Windows system running Windows 7 and it acted as the server.

4.2 Evaluation and results

We created five different instances of attack classes which included Pyloris, LOIC,XOIC, SynGUI and DoSIM attacks for testing the proposed classifier. The dataset was prepared in such a way that a total of approximately 5000 attack instances were collected. The total instances were split into multiple datasets of 100, 500, 1000, 2000 and 5000 instances respectively. Dataset 'A' has 100 instances of all the above attack types. Dataset 'B' has 500 instances and so on. In order to measure the accuracy of the proposed classifier, we have used standard detection measures. We refer the proposed work with Recall, Precision and F-Measure. They are defined as shown in Eqs. (7), (8) and (9).

$$\text{Recall Rate} = \text{True Positive}/(\text{True Positive} + \text{False Negative}) \times 100 \quad (7)$$

$$\text{Precision} = \text{True Positive}/(\text{True Positive} + \text{False Positive}) \times 100 \quad (8)$$

$$\text{F - Measure} = 2 * (\text{Precision} * \text{Recall})/(\text{Precision} + \text{Recall}) \quad (9)$$

where True Positive is given as attack instances that were rightly identified as attacks. False Positive being genuine request or traffic wrongly interpreted as an attack. True Negative is where the genuine request or traffic is flagged as genuine traffic and False Negative being malign request or attack traffic that were wrongly classified as genuine traffic. Recall measure is also known as True Positive Rate. It can be defined as malign requests or malign traffic being admitted as malign traffic itself. Precision can be said as the rate of proportion of True positives with respect to both true positive rate and false positive rates combined. The measurements discussed above are summarised in Table 7 using confusion matrix.

False positive rate acts as an important measure against other security threats such as phishing webpage detection, medical data mining, intrusion detection etc. When the false positive rate is lower, it translates to better classifier performance. The graph comparing the false positive rate of different dataset combinations is given in Fig. 3.

During the experiments, we measured the number of packets arriving per second in the virtual cloud network. We have summarised the TCP and UDP packet arrival rate

Table 7 Confusion matrix

Test measures	Genuine traffic	Malign traffic	Output measure
Positive	False positive	True positive	Precision
Negative	True negative	False negative	–
Output measure	Specificity	Recall	–

with increasing number of attackers in Table 8. The table indicates how the number of TCP and UDP packets being sent increases quickly with respect to increase in the number of attackers. We can see the number of attackers influence the number of malign packets sent over the network. We have used the tool IPTraf to automatically calculate the total number of TCP and UDP packets sent.

Recall, Precision and F-Measure values obtained for the different dataset sizes are givenin Table 9.

The average Recall, Precision and the F-Measure values obtained in the experiments is given in Table 10.

Figure 4 shows the accuracy analysis achieved without feature selection for various classification algorithms namely Naïve Bayes, Decision Tree Algorithm (C4.5), Multilayer Perceptron from Artificial Neural Networks (ANN), Support Vector Machine (SVM) and IRCS.

From Fig. 4, it is observed that the accuracy of the proposed system IRCS is high when compared to the other existing works namely Naïve Bayes, C4.5, ANN, SVM. This achievement is obtained due to the use of rules and also the effective application of experts opinions in IRCS.

$$\text{Accuracy} = \left\{ \frac{(\text{True Positive} + \text{True Negative})}{(\text{True Positive} + \text{True Negative} + \text{False Positive} + \text{False Negative})} \right\} * 100 \quad (10)$$

We calculated the accuracy of the proposed classifier and then compared it with similar literature works. Accuracy can be defined as Eq. (10).

The accuracy rate obtained by the classifier for different datasets is given in Table 11. We can see that the classifier performs better when it can study large attack instances. The proposed IRCS has higher accuracy rates as the size of the dataset increases.

The accuracy rate obtained by the proposed classifier along with the accuracy rates achieved by previous literature works [29] is shown in Table 12 and Fig. 4.

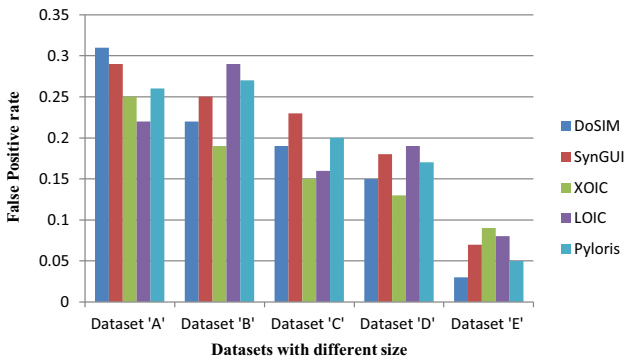

Fig. 3 False positive rates of the proposed classifier with different dataset sizes

Figure 5 shows the accuracy analysis achieved with feature selection for various classification algorithms namely Naïve Bayes, C4.5, ANN, SVM and the proposed IRCS.

From Fig. 5, it is observed that the accuracy of the proposed system IRCS is high when compared to the other existing works namely Naïve Bayes, C4.5, ANN and SVM. This achievement is obtained due to the use of rules, the effective application of experts opinions and with feature selection methods in IRCS.

5 Conclusion and future works

Cloud network is a model for data storage and communication has been used in various organisations and industries. These organisational data are also exposed to large volume of attacks specifically targeting their cloud network resources. Therefore, an intelligent security system is proposed in this paper which uses newly proposed feature selection and classification techniques for effective detection of DoS attacks. Moreover, the proposed model uses the experts advice for making final decisions on security and hence the proposed system achieved an accuracy of 98.5% with respect to the detection of attacks. The experimental results obtained from this work showed that

Table 8 TCP packet arrival rate

No. of attackers	No. of TCP packets per second	No. of UDP packets per second
2 Virtual machines	933	415
3 Virtual machines	2894	1843
4 Virtual machines	89,582	77,951
5 Virtual machines	132,499	127,562
6 Virtual machines	206,647	193,195

Table 9 Recall, Precision and F-Measure values obtained

Measure	DoSIM	SynGUI	XOIC	LOIC	Pyloris
Dataset 'A' (100 instances)					
Recall	0.76	0.82	0.86	0.8	0.8
Precision	0.79	0.74	0.82	0.82	0.8
F-Measure	0.78	0.78	0.83	0.86	0.77
Dataset 'B' (500 instances)					
Recall	0.82	0.8	0.81	0.86	0.84
Precision	0.8	0.82	0.82	0.82	0.83
F-Measure	0.81	0.86	0.81	0.82	0.84
Dataset 'C' (1000 instances)					
Recall	0.8	0.85	0.86	0.85	0.86
Precision	0.86	0.85	0.83	0.84	0.87
F-Measure	0.83	0.85	0.85	0.84	0.83
Dataset 'D' (2000 instances)					
Recall	0.93	0.93	0.92	0.94	0.9
Precision	0.936	0.92	0.93	0.91	0.938
F-Measure	0.93	0.92	0.91	0.93	0.916
Dataset 'E' (5000 instances)					
Recall	0.98	0.97	0.96	0.95	0.941
Precision	0.98	0.989	0.97	0.97	0.98
F-Measure	0.981	0.976	0.957	0.988	0.966

Table 10 Average Recall, Precision and F-Measure values

Dataset	Recall	Precision	F-Measure
'A' (100 instances)	0.81	0.80	0.80
'B' (500 instances)	0.83	0.82	0.83
'C' (1000 instances)	0.84	0.85	0.84
'D' (2000 instances)	0.92	0.93	0.92
'E' (5000 instances)	0.96	0.98	0.98

Table 11 Accuracy rate of IRCS for different datasets

Dataset	Accuracy of IRCS (%)
Dataset 'A' (100 instances)	81.90
Dataset 'B' (500 instances)	85.37
Dataset 'C' (1000 instances)	89.55
Dataset 'D' (2000 instances)	93.62
Dataset 'E' (5000 instances)	98.53

the knowledge provided by a domain expert increases the efficiency of the proposed classifier when it is compared with the commonly used classification algorithms. The major advantages of the proposed model include the reduction in false positive rate and increase in security.

Table 12 Comparison of IRCS with existing works

Work based upon classification technique	Accuracy rates achieved without feature selection (%)	Accuracy rates achieved with feature selection
Naïve Bayes	64.53	68.53
Multilayer perceptron	90.33	94.33
SVM	92.27	96.27
Decision tree	71.71	74.71
IRCS	96.5	98.5

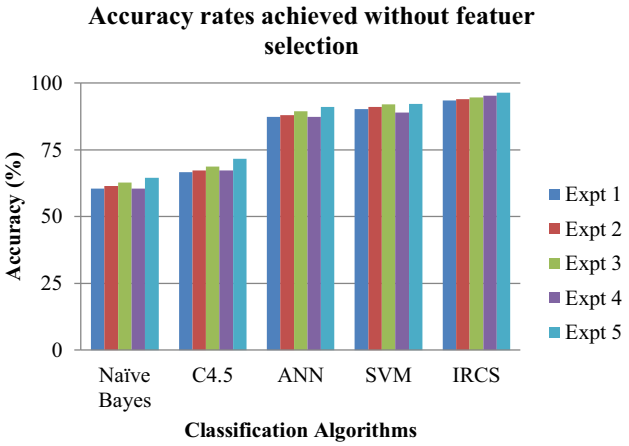

Fig. 4 Accuracy analysis achieved without feature selection for classification algorithms

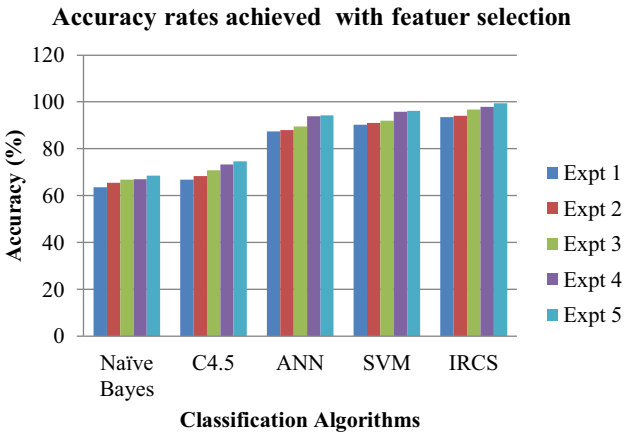

Fig. 5 Accuracy rates achieved with feature selection for classification algorithms

Future works in this direction can be the use of temporal constraints to capture dynamic nature of attacks. Moreover, fuzzy rules can be used to handle uncertainty and to make further accurate decisions.

References

1. Khorshed, M.T., Shawkat Ali, A.B.M., Wasimi, S.A.: Classifying different denial-of-service attacks in cloud computing using rule-based learning. Secur. Commun. Netw. **5**(11), 1235–1247 (2012)
2. Ficco, M., Rak, M.: Stealthy denial of service strategy in cloud computing. IEEE Trans. Cloud Comput. **3**(1), 80–94 (2015)
3. Ganapathy, S., Kulothungan, K., Muthurajkumar, S., Vijayalakshmi, M., Yogesh, P., Kannan, A.: Intelligent feature selection and classification techniques for intrusion detection in networks: a survey. EURASIP J. Wirel. Commun. Netw. **271**(1), 1–16 (2013)
4. Arul Xavier, V.M., Annadurai, S.: Chaotic social spider algorithm for load balance aware task scheduling in cloud computing. Clust. Comput. (2018). https://doi.org/10.1007/s10586-018-1823-x
5. Muthurajkumar, S., Ganapathy, S., Vijayalakshmi, M., Kannan, A.: Secured temporal log management techniques for cloud. Procedia Comput. Sci. **46**, 589–595 (2015)
6. Ren, Y., Wang, J., Feng, X., Younn, G., Kim, J.-U.: A hierarchical clustering based method to evaluate reuse of rare earth tailings under cloud computing environment. Clust. Comput. (2018). https://doi.org/10.1007/s10586-017-1654-1
7. Patel, A., Taghavi, M., Bakhtiyari, K., Júnior, J.C.: An intrusion detection and prevention system in cloud computing: a systematic review". J. Netw. Comput. Appl. **36**(1), 25–41 (2013)
8. Li, Z., Sun, W., Wang, L.: A neural network based distributed intrusion detection system on cloud platform. In: Proceedings of 2nd IEEE Conference on Cloud Computing and Intelligence Systems, pp. 75–79 (2012)
9. Choi, J., Choi, C., Ko, B., Kim, P.: A method of DDoS attack detection using HTTP packet pattern and rule engine in cloud computing environment. Soft. Comput. **18**(9), 1697–1703 (2014)
10. Pradeepthi, K.V., Kannan, A.: Cloud attack detection with intelligent rules. KSII Trans. Internet Inf. Syst. **9**(10), 4204–4221 (2015)
11. Kim, H.-Y.: An energy-efficient load balancing scheme to extend lifetime in wireless sensor networks. J. Clust. Comput. **19**, 279–283 (2016)
12. Logambigai, R., Kannan, A.: Fuzzy logic based unequal clustering for wireless sensor networks. Wirel. Netw. **22**, 945–957 (2016)
13. Ruby D, Vijayalakshmi M, Kannan A: Intelligent relay selection and spectrum sharing techniques for cognitive radio networks. J. Clust. Comput., pp. 1–12 (2017)
14. Gupta, S., Kumar, P., Abraham, A.: A profile based network intrusion detection and prevention system for securing cloud environment. Int. J. Distrib. Sens. Netw., pp. 1–12 (2013)
15. Santhosh Kumar, S.V.N., Yogesh, P.: Energy efficient and secured distributed data dissemination using hop by hop authentication in WSN. Wirel. Netw. (2017)
16. Jansen, W.A.: Cloud hooks: security and privacy issues in cloud computing. In: Proceedings of 44th Hawaii International Conference on System Sciences, 1 Jan 2011
17. Yu, S., Tian, Y., Guo, S., Wu, D.O.: Can we beat DDoS attacks in clouds? IEEE Trans. Parallel Distrib. Syst. **25**(9), 2245–2254 (2014)
18. Girma, A., Garuba, M., Li, J., Liu, C.: Analysis of DDoS attacks and an introduction of a hybrid statistical model to detect DDoS attacks on cloud computing environment. In: Proceedings of 12th International Conference on Information Technology-New Generations, pp. 212–217, 13–15 Apr 2015
19. Arshad, J., Townend, P., Xu, J.: A novel intrusion severity analysis approach for clouds. Future Gener. Comput. Syst. **29**(1), 416–428 (2013)
20. Selvi, M., Logambigai, R., Ganapathy, S., Khanna Nehemiah, H., Kannan, A.: An intelligent agent and FSO based efficient routing algorithm for wireless sensor network. In: Second International Conference on Recent Trends and Challenges in Computational Models (ICRTCCM), IEEE, pp. 100–105 (2017)
21. Selvi, M., Logambigai, R., Ganapathy, S., Sai Ramesh, L., Khanna Nehemiah, H., Kannan, A.: Fuzzy temporal approach for energy efficient routing in WSN. In: Proceedings of the International Conference on Informatics and Analytics, ACM, pp. 1–5 (2016)
22. Selvi, M., Nandhini, C., Thangaramya, K., Kulothungan, K., Kannan, A.: HBO based clustering and energy optimized routing algorithm for WSN. In: Eighth International Conference on Advanced Computing (ICoAC), IEEE, pp. 89–92 (2016)
23. Selvi, M., Velvizhy, P., Ganapathy, S., Khanna Nehemiah, H., Kannan, A.: A rule based delay constrained energy efficient routing technique for wireless sensor networks. J. Clust. Comput. (2017). https://doi.org/10.1007/s10586-017-1191-y
24. Munuswamy, S., Saravanakumar, J.M., Sannasi, G., Harichandran, K.N., Arputharaj, K.: Virtual force-based intelligent clustering for energy-efficient routing in mobile wireless sensor networks. Turk. J. Electr. Eng. Comput. Sci. (2017). https://doi.org/10.3906/elk-1706-226
25. Sindhu, S.S.S., Geetha, S., Kannan, A.: Decision tree based light weight intrusion detection using a wrapper approach. Exp. Syst. Appl. **39**(1), 129–141 (2012)
26. Chonka, A., Xiang, Y., Zhou, W., Bonti, A.: Cloud security defence to protect cloud computing against HTTP-DoS and XML-DoS attacks. J. Netw. Comput. Appl. **34**(4), 1097–1107 (2011)
27. Wu, Y., Tseng, H., Yang, W., Jan, R.: DDoS detection and traceback with decision tree and grey relational analysis. In: Third International Conference on Multimedia and Ubiquitous Engineering (MUE), Qingdao, China, pp. 306–314, 4–6 June 2009 (2009)
28. Datasets: KDDCUP 1999 data. In: The Fifth International Conference on Knowledge Discovery and Data Mining 1999. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
29. Oktay, U., Sahingoz, O.K.: Attack types and intrusion detection systems in cloud computing. In: Proceedings of 6th International Conference on Information Security & Cryptology, pp. 71–76, 23-24 May 2013

Rakesh Rajendran is a full time Ph.D. research scholar in the Department of Information Science and Technology, College of Engineering Guindy, Anna University, Chennai, India. He completed his B.E. and M.E. degrees in CSE from Anna University, Chennai, India. He has published 5 research papers in international journals and conference proceedings to his credit. His areas of interests include Web Security, Cloud Computing and Soft Computing.

S. V. N. Santhosh Kumar is a full time Ph.D. research scholar of the Department of Information Science and Technology, College of Engineering Guindy, Anna University, Chennai, India. He works in the areas of security and data dissemination in wireless sensor networks. His areas of interests include Wireless Sensor Networks, Internet of Things, Mobile Computing. He received his B.E. degree in Computer Science and Engineering and M.E. degree in Software Engineering from Anna University, Chennai, India in the years 2011 and 2013 respectively.

Yogesh Palanichamy is working as an Associate Professor in the department of Information Science and Technology, College of Engineering Guindy, Anna University, Chennai, India. He received his B.E. and M.E. degrees from Madurai Kamaraj University, India and Ph.D. degree from Anna University, Chennai, India. His areas of interests include Wireless Sensor Networks, Internet of Things, Mobile Computing and Multimedia Communication. He has published more than 40 papers. He has produced nine Ph.D. research scholars and eight research scholars are currently pursuing Ph.D. programme under his supervision.

Kannan Arputharaj is currently working as a Professor in the Department of Information Science and Technology, College of Engineering, Anna University, Chennai, India. He has 8 years of experience in software development at Bhabha Atomic Research Centre, Mumbai, India and 24 years of teaching at College of Engineering Guindy, Anna University, Chennai, India. He has published more than 300 research papers in reputed journals and conference proceedings. His areas of interests include Database Management System, Networks, Soft Computing and Artificial Intelligence.