Cloud Attack Detection with Intelligent Rules

Pradeepthi K.V[1] and Kannan A[1]
[1] Department of Information Science and Technology, College of Engineering, Guindy,
Anna University,
Chennai-600025, India
[e-mail: pradeepthi.kv@gmail.com, kannan@annauniv.edu]
*Corresponding author: Pradeepthi K.V

Received March 26, 2015; revised June 17, 2015; accepted July 2, 2015;
published October 31, 2015

Abstract

Cloud is the latest buzz word in the internet community among developers, consumers and security researchers. There have been many attacks on the cloud in the recent past where the services got interrupted and consumer privacy has been compromised. Denial of Service (DoS) attacks effect the service availability to the genuine user. Customers are paying to use the cloud, so enhancing the availability of services is a paramount task for the service provider. In the presence of DoS attacks, the availability is reduced drastically. Such attacks must be detected and prevented as early as possible and the power of computational approaches can be used to do so. In the literature, machine learning techniques have been used to detect the presence of attacks. In this paper, a novel approach is proposed, where intelligent rule based feature selection and classification are performed for DoS attack detection in the cloud. The performance of the proposed system has been evaluated on an experimental cloud set up with real time DoS tools. It was observed that the proposed system achieved an accuracy of 98.46% on the experimental data for 10,000 instances with 10 fold cross-validation. By using this methodology, the service providers will be able to provide a more secure cloud environment to the customers.

Keywords: Cloud, DoS attack, rule-base, feature selection, classification, expert systems.

http://dx.doi.org/10.3837/tiis.2015.10.025 **ISSN : 1976-7277**

1. Introduction

The advent of cloud has revolutionized the web application domain. Any user with a computer and an internet connection can use the resources like software or infrastructure placed elsewhere in the cloud. This has helped students, researchers and entrepreneurs to use as per their need and reduce cost. All major software companies like Amazon, Google, etc have joined cloud computing arena and initiated their own cloud services, which are extremely popular. The cloud data centres are also playing an important role in enhacing the data storage for mobile networks [1].

With the advances in cloud computing, the negative aspects of usage are also increasing. The news of hacking or data leakages in cloud is on the rise these days. In the current scenario, the availability of the cloud and security of customer data in the cloud is of utmost importance for the customers as well as to the cloud service providers. Cloud providers take all the security measures and put up the required firewalls to prevent attacks. But still the attacks are happening because there are certain ports like port 80 which has to be left open for the consumer traffic. The attackers make use of these open ports that have been left open for the consumer traffic to send flood pings and make the system unavailable for a genuine consumer. This becomes the Denial of Service (DoS) attack. Some common denial of service attacks are: UPD attack, HTTP attack, ICMP ping flood, Slowloris, SYN flood, ping of death, etc. The recognition of the pattern during an attack is of at most importance, as an attack should be terminated in the early stage itself and so that the magnitude of destruction it can cause can be minimized.

To solve the problem of attack detection in cloud, Artificial Intelligence (AI) techniques can be used because AI has the ability to solve a problem after learning from certain examples. Classification methods have been applied to the field of attack detection in many papers. In this paper, we propose a new rule based expert system to solve this problem. An important pre-processing step that is often performed before classification is feature selection. Features selection is applied when there are many features and the dataset size is large. This reduces the time required to perform the classification. In this paper, the detection of DoS attacks in cloud is done through a knowledge based feature selection method and the design of an intelligent rule based classification system. The knowledge based feature selection uses a novel entity called weight, which is assigned by the domain expert while training the dataset with neural networks. This weight along with the information gain factor is used for the purpose of feature selection. There are many techniques like information gain, entropy, gain ratio, etc which are being currently used. But they do not give any weight to the domain based importance of a feature while doing a feature selection. Hence, this methodology of knowledge based feature selection has been proposed for promoting a feature which would be able to aid classification of an instance based on neural networks and the domain expert opinion. The dataset is first trained using a back propagation neural network and then rules formulated by an intelligent rule based system based on expert weight. It can be inferred from our results that the accuracy improves by weight adjustment using the domain experts knowledge.

The reminder of this paper is organized as follows: Section 2 gives a literature survey of the related works; Section 3 describes the proposed system in detail, while Section 4 talks about the experimental setup done, Section 5 gives the ensuing results and related discussion and finally Section 6 concludes the paper.

2. Literature Survey

Cloud computing is a vast evolving field and the need for cloud security research is more important because of the increasing attack instances. There are many works in the literature on security and privacy of data. Most important among these techniques are key management [2], admission control [3] and intrusion detection [4], [5]. A survey of the different client side and server side protection mechanisms available is discussed in [6], where as the technical issues in cloud security from the browser and web service side have been touched up on in [7]. Information leakage in third party compute clouds through VM side channel attacks is discussed and mitigating techniques have been shared in [8].

Many researchers have utilized the unique features of the cloud, like its dynamic research allocation [9], software defined networking [10], statistical modeling [11] to handle attack scenarios. In [12], DoS attack on the Google cloud and the effects it has on the servers is discussed. The authors have expressed that the current protection mechanisms are only a temporary solution. Different attack types that affect the cloud performance were studied in [13], [14], [15] and solutions like graphical model [13], taxonomy [14], ellipsoid and kernel component analysis [16] has been proposed for attack detection.

Machine learning is being used quite efficiently in many cloud security related research works. Insider activity in the cloud can be monitored through performance data. Rule based classification was done to classify the insider activity into different types in [17]. Different classification algorithms like Naïve Bayes, Multilayer Perceptron, SVM, Decision tree and PART were used and their results were compared. In another work [18], a cloud trace back methodology has been proposed which is said to trace back the source of a HTTP denial of service attack or a HTML denial of service attack. The detection and filtering of these attacks has been done by using back propagation neural networks.

Severity analysis for intrusion in cloud is proposed in [19], where the virtual machine parameters have been analyzed. Then using machine learning techniques the severity of the intrusion is predicted. The different intrusions hampering the integrity, confidentiality and availability of the different cloud services have been surveyes in [20]. Intrusion detection in a network by applying conditional random fields and layered approach was done in [21]. They have used the concept of selecting features manually, instead of applying the automatically selected features from feature selection algorithms and shown that manually selected features give better performance. Machine learning has been incorporated in intrusion detection systems[22] by using many techniques like wrapper approach [23], learning model [24] and particle swarm approach [25].

The most prevalent attacks in cloud are denial of service attacks, cross side channel attacks in the virtual machine, phishing, shared memory attacks and insider malicious activities. A cloud set up was build and the attack scenarios were recreated with 5000 instances and 8 attributes and performed classification with SVM by Tanzim et al[26]. The same authors in another paper [27] have used 14 attributes and 536 instances and done the classification of the different denial of service attacks on a cloud environment. They have used machine learning algorithms to classify the attacks. In the current scenario were the processor can deal with any number of instances in seconds, more data can be generated and used to classify the system. To achieve a better accuracy in detection of attack, we have replaced the generic classification algorithms with a more problem specific intelligent rule based expert system to do the classification, as it has been proved time and again that applying domain knowledge to computational approach improves the results to a greater extent.

Using the facts and rules conceived, we have formulated an intelligent rule based system that can effectively classify the given data into the different attack types. Another important aspect of this paper is that an experimental cloud setup was done to generate a dataset of 10,000 instances.

3. Proposed Work

As discussed earlier, firewalls are unable to detect DoS attack. Cloud is loaded with unlimited resources and is dynamic in nature. But still, when the services of a particular service provider are targeted, then the genuine user gets affected and the services that need to be provided to them are compromised.

In this section, a detailed explanation of the proposed methodology is given. The two techniques proposed in this paper are features selection and classification with intelligent rule based system. In this work, a cloud environment was setup to conduct all the DoS attacks and understand their effects on the cloud. The performance parameters were observed and a dataset was constituted. Using this extensive dataset, all the further analysis has been carried out. The existing works in this area take up the classification problem and use the existing classification algorithms to perform the classification. Classification algorithms do not use domain knowledge to solve a problem. In this paper, we use an intelligent rule based system to enhance the classification accuracy compared to the normal classification algorithms by making use of the domain knowledge. The intelligent rule based classification method consists of a rule base for firing rules and perform deductive inference. The following subsections discuss in more detail the feature selection and classification modules.

3.1 System Architecture

The architecture of the intelligent rule based classifier is shown in **Fig. 1**. The attack generation module uses tools like LOIC, SynGUI, ping flood, Unicorn and Pyloris for simulating the attacks. When the cloud comes under a DoS attack, the cloud parameters like the various CPU, memory, network and storage values get affected. The performance parameters during these attacks and even during the no attack phase are monitored, so that we can differentiate between an attack and a normal acceptable cloud behavior. The performance metric capture during all the different attack phases and the no attack phase, form the dataset for further detection. This data now undergoes feature selection through the novel knowledge based feature selection technique. Feature Selection is performed on all the features to reduce their number and select only those that will provide a higher accuracy. The selected features are then used to classify the different types of attacks using the classification module. The classification module works based on the intelligent rule based system.

The Intelligent rule based system, uses the inference engine to select the rules and schedule them to perform forward chaining inference. Moreover, the interpreter present in the inference engine carries out the tasks of rule matching and rule execution to perform deductive inference. Based on the results of the classifier; the decision manager module decides on the further course of action with the help of the intelligent rule based system. If the classifier detects the presence of an attack, then the particular process would be terminated, to prevent the further spread and depletion of the cloud resources.

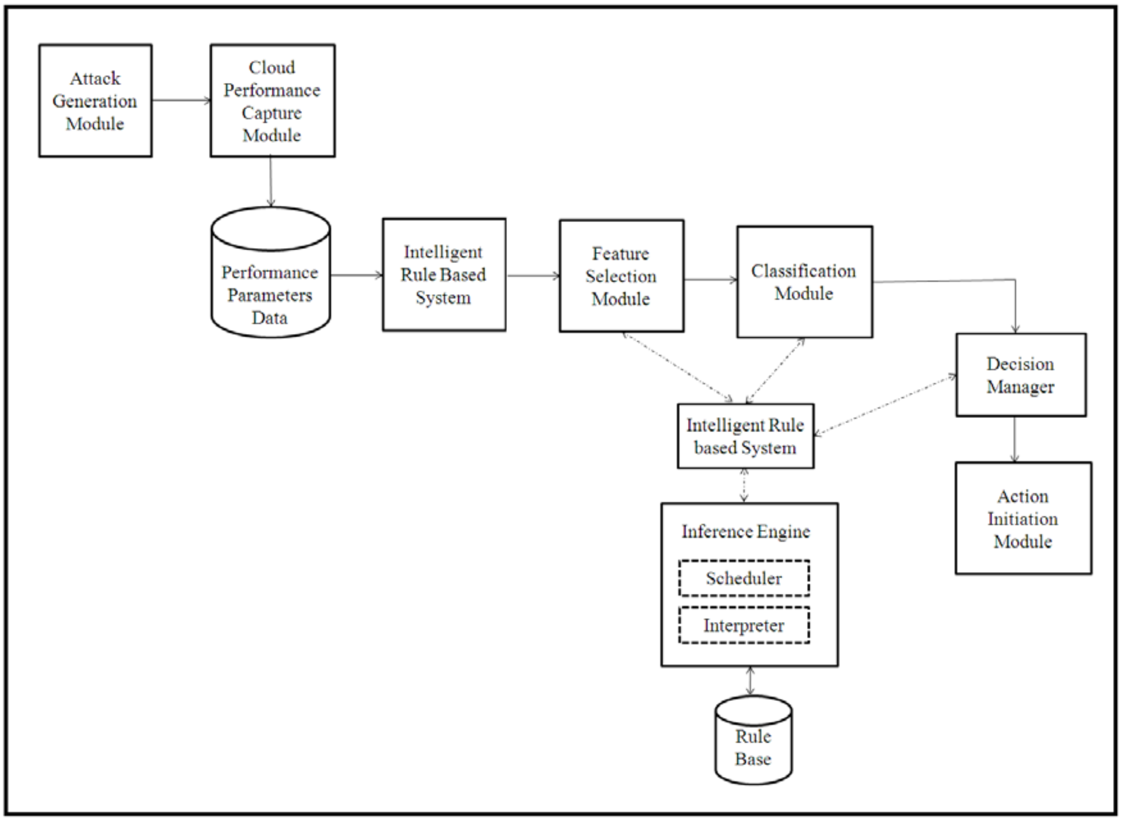


Fig. 1. Intelligent Rule based system- Architecture

3.2 Feature selection

The different features in memory, process, disk and network which get significantly affected during an attack were observed. There are 47 features in the dataset. After carefully studying the different features and applying the existing feature selection techniques, as expected, the accuracy was found to increase and the classification time got reduced, as the dataset size decreases. While applying the existing feature selection techniques, it was noticed that some features which are relevant to one or two of the different attacks classes was being missed in the reduced feature list. Applying mathematical formulae like information gain or entropy, it can be seen that at times the set of features selected and their order of preference is not always as perceived by a domain expert. By allowing a domain based knowledge system to select the features, we are able to avoid certain discrepancies that the automatic methods could make. The dominant features essential for the accurate classification of attacks get selected in our methodology by the intelligent rule based system. There are 5 classes into which the data has to be classified.

Let C1, C2, C3, C4 and C5 be the 5 classes. The probability that a random instance ‘a’ belongs to a particular class Ci is

$$Ci/Ci+(C1+\ldots+C5), \qquad (1)$$

where i ranges from 1 to 5 in our case.

Let us assume there are only two classes of data for classification say p1 and p2. Then the information needed to extract the result is given by

$$i(p_1, p_2) = -\frac{p_1}{p_1+p_2}\log_2\frac{p_1}{p_1+p_2} - \frac{p_2}{p_1+p_2}\log_2\frac{p_2}{p_1+p_2} \qquad (2)$$

If we generalize the information gain to prediction the class of an instance then it can be given as

$$\text{Information Gain for a instance, } i = -\sum_{i=0}^{n} p_i \log_2(p_i) \text{[28]} \qquad (3)$$

where pi is the probability that the instance belong to class Ci.

We conceive a new parameter called weight that can be assigned to every attribute. It is called W1. It is formulated as follows,

$$W_1 = \sum_{i=1}^{n} \frac{Information\ Gain(i) + Expert\ weight\ (i)}{2n} \qquad (4)$$

Where expert weight (i) is assigned to an attribute by a domain expert and n is the number of instances.

$$W_2 = Information\ Gain\,(i) \qquad (5)$$

where the information gain for a feature is calculated as follows.
\

Let us consider a threshold τ. The weight of a feature should not be more than the threshold τ.

$$Weight\ W = \begin{cases} W_1 + Correction, & W_1 - W_2 < \tau \\ W_1 - Correction, & W_1 - W_2 \geq \tau \end{cases} \qquad (6)$$

Optimal values for threshold and correction were chosen based on experiments.

Algorithm for Intelligent rule based feature selection:

Let f_i= feature i(i= 1 to n(total number of features)), w_e= weight assigned by expert and w_i= weight of feature f_i.

Input: Complete feature set F_C
Output: Reduced feature set F_R

```
$\forall$fi, wi=0                            // initialize the weight wi as 0 for all the feature.
for $\forall$fi $\in$ FC,                      // for all fi in FC, a loop is instantiated.
        if we>threshold T                  // if the expert weight of a feature is more than T.
        then wi=wi+1,                      // then the feature's weight is incremented by 1.
        if wi> T, then FR= fi $\cup$ FR       // if the feature weight is greater than T then it is
                                              added to the reduced feature set FR.
i=i+1                                      // increment loop variable to get the next feature
                                              in FC.
```

The threshold for determining the selected weights is determined by the domain experts. The ideology of manual feature selection [21] using the knowledge base and has been applied in our proposed system and it is observed that it gives us the freedom of picking features which are going to effectively determine the attack classification. We observed increase in the accuracy. Using knowledge based manual feature selection, 30 features were selected for the purpose of the attack classification.

3.3 Classification

For the purpose of classification, the intelligent rule based classifier is used. The rules of the system have been set by using thresholds determined by knowledge base. We are dealing with a multi-class classification problem here; hence the complexity of the system is more. In a normal decision tree algorithm, the criteria for classification are selected using entropy, gini impurity index, and information gain or variance reduction. If the selection criteria are entropy then the smallest value is selected and if it is information gain, the maximum value is selected for making the decision. Our concept is similar to the decision tree methodology in ID3 or C4.5 algorithm but the selection criterion for the decision rules formation is different. Instead of using either the entropy or the information gain, each attribute is assigned an initial weight based on an expert weight. Based on the expert weight assigned to the attribute, the classification is carried out. The expert weights are assigned by the knowledge base. The prerequisite for the whole process is the formation of a knowledge base with the initial rules. For forming these initial rules, the dataset is studied and the expert weights are assigned to the attributes. The assignment of expert weights is done in such a fashion that the attributes that have high expectation for effectively classifying the classes are assigned higher weights and vice versa. Moreover, the dataset is trained with a back-propagation neural network and the rules are finalized by adjusting the weights. Such rules are compared with the rules formed by the domain experts. Finally, the matching rules are identified and are stored in the knowledge base. Rule matching is performed by building discriminant network and forward chaining inference method is used to perform deductive inference.

Each instance has a value for n attributes. It can be represented as

$$instance\ a = \{a_1, a_2, \dots a_n\} \tag{7}$$

and it can be assigned to a distinct class Ci.

$$a \rightarrow C_i \tag{8}$$

Generic if- then rules used in our system are

$$If\ p_1 is\ condition_1, \dots p_n\ is\ condition_n\ then\ Class\ is\ C_1 \tag{9}$$

Where p1 ….pn are the parameters and C1 is a class to which the instance belongs.

Algorithm for Classification performed by the intelligent rule based system is given below.

```
Let fi= feature i(i= 1 to n),
wi= weight of feature fi.

Input: Unclassified instances
Output: Classified instances

        // calculation for the weight $w_i$
        $\forall f_i,\ w_i = \begin{cases} w_1 + \text{Correction}, & w_1 - w_2 < \tau \\ w_1 - \text{Correction}, & w_1 - w_2 \geq \tau \end{cases}$
        where $w_1 = \sum_{i=1}^{n} \frac{\text{Information Gain(i)+Expert weight(i)}}{2n}$ and $w_2 = \text{Information Gain(i)}$.
```

```
     // assignment of root node
2    Root Node= $f_i$ with max($w_i$)
     // assignment of leaf node
3    If all leaves at the level have same class, then
4    Return.
5    Else
6    Child Node= $f_i$ with max($w_i$)
7    Repeat Step 3.
```

Gist of the rules conceived for the knowledge base is given below.

```
i) For the no attack scenario,
the rules are
        if Process blk>pbmin,
         load-1m>load1min and <load1max,
         load-5m>load5min and <load5max,
         SDButil>SDBmin and <0
Then attack type="No attack"

ii) For LOIC-TCP attack,
the rules are
        if CPUusr>CPUusrmin and <CPUusrmax,
         CPU sys >CPUsysmin and < CPUsysmax,
         load-1m >load1min and < load1max,
         mem used>memmin and <memmax,
        mem buff>mem_buffmin and<mem_buffmax,
        mem free>mem_freemin and <mem_freemax,
         Process run>p_run min and <p_runmax,
        Process blk> pbmin and < pbmax,
        TCP time > TCP_tmin and < TCP_tmax,
        SDButil > SDBmin and< SDBmax,
         VM majpf>VMmajmin and < VMmajmax,
        VM minpf> VMminmin and < VMminmax
Then attack type="LOIC-TCP attack"
iii)For Pyloris attack,
 the rules are
        if Network recv>Nwrcvmin and < Nwrcvmax,
        CPUusr > CPUusrmin and <CPUusrmax,
        CPUsys> CPUsysmin and < CPUsysmax,
         CPUwait> CPUwmin and < CPUwmax,
        Disk read>Dreadmin and <Dreadmax,
         Disk write> Dwritemin and <Dwritemax,
        load-1m> load1min and < load1max,
         load-5m> load5min and <load5max,
         load-15m>load15min and <load15max,
         Process run >p_runmin and <p_runmax,
        Process blk >pbmin and <pbmax,
        IO read >IOreadmin and <IOreadmax,
        IO write>IO write min and <IOwrite max,
        frg>frgmin and <frgmax,
        TCP syn >TCP_synmin and < TCP_synmax,
         VM majpf> VMmajmin and < VMmajmax
Then attack type="Pyloris attack"
```

```
iv) For Unicorn attack,
the rules are
        if Process blk>pbmin and < pbmax,
        TCP time > TCP_tmin and < TCP_tmax,
        TCP Clo> TCP_Cmin and < TCP_Cmax
Then attack type=”Unicorn attack”

v) For SynGUI attack,
 the rules are
        if Network recv>Nwrcvmin and < Nwrcvmax,
         CPUsys>CPUsysmin and <CPUsysmax,
        CPUwait> CPUwmin and < CPUwmax,
        CPUsiq >CPUsiqmin and < CPUsiqmax,
        Disk read>Dreadmin and <Dreadmax,
        Process blk>pbmin and < pbmax,
        IO read >IOreadmin and <IOreadmax
Then attack type=”Syn GUI attack”

vi) For Ping Flood attack,
the rules are
        if Network recv>Nwrcvmin and < Nwrcvmax,
         CPUwait> CPUwmin and < CPUwmax,
        CPUsiq >CPUsiqmin and < CPUsiqmax,
        Process new> p_newmin and <p_newmax,
         System CSW>CWS_sysmin and < CWS_sysmax,
        sockets raw>socket_rawmin and < socket_rawmax
Then attack type=”Ping Flood attack”
```

The calculation for evaluating the minimum and maximum values of the different parameters is shown below. For Process blk (pbmin), the minimum value is

$$pb_{min} = min\{pb_1, pb_2, pb_3, \dots, pb_n\} \tag{10}$$

where m is the number of instances in a particular experiment.

For calculating the maximum value of a parameter, for example load-1m,

$$load1m_{max} = max\{load_1, load_2, load_3, \dots, load_n\} \tag{11}$$

where n is the number of instances in a particular experiment.

The list of parameters names shown and their corresponding features names are listed in the appendix. The minimum and maximum threshold values for these features were set based on a series of experiments that were conducted with the individual attack tools under different system conditions, like varying the load on the memory, processor, network, etc.,

New attacks can also be detected based on these rules as the knowledge base is constantly updated based on newly added instances and the rules get refined. In the results section, we have compared the results of our classifier with Decision Tree and also done a comparison of the accuracy with other works mentioned in the literature.

4. Experimental Setup

Two machines were taken for this experimental purpose. They had the following configuration: Intel Core i7 with dual processors of 3.40GHz with 4GB RAM .The cloud was setup on one machine and the attack tools were run from the other machine. The cloud machine had Cent OS with Eucalyptus cloud installed in it. The attack machine had Windows OS, on which the various attack tools were run.

In our experiment, we attack the cloud with DoS tools. DoS attacks create futile traffic packets and send them to the target system, so that the genuine customers will not be able to make use of the resources. The different attack tools that are being used are LOIC, Pyloris, Unicorn, SynGUI and pingflood. These are DoS tools which work in such a way that CPU usage, memory and network bandwidth of the cloud gets depleted. Hence, the genuine users are denied the opportunity to utilize the cloud resources which they are entitled to. Fig 2 shows the screen shot of the cloud system performance during the no attack phase.

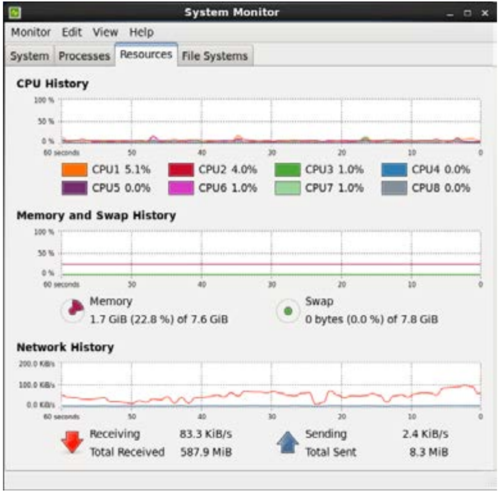

System Monitor
Monitor Edit View Help
System Processes Resources File Systems
CPU History
100 %
50 %
0 %
60 seconds 50 40 30 20 10 0
CPU1 5.1% CPU2 4.0% CPU3 1.0% CPU4 0.0%
CPU5 0.0% CPU6 1.0% CPU7 1.0% CPU8 0.0%
Memory and Swap History
100 %
50 %
0 %
60 seconds 50 40 30 20 10 0
Memory
1.7 GiB (22.8 %) of 7.6 GiB
Swap
0 bytes (0.0 %) of 7.8 GiB
Network History
200.0 KiB/s
100.0 KiB/s
0.0 KiB/s
60 seconds 50 40 30 20 10 0
Receiving 83.3 KiB/s
Total Received 587.9 MiB
Sending 2.4 KiB/s
Total Sent 8.3 MiB

Fig. 2. System parameters- No attack phase.

Fig 3 shows the scenario when a Unicorn DoS tool in its full force is attacking the cloud. There is immense increase in the network traffic and the CPU utilization also fluctuates.

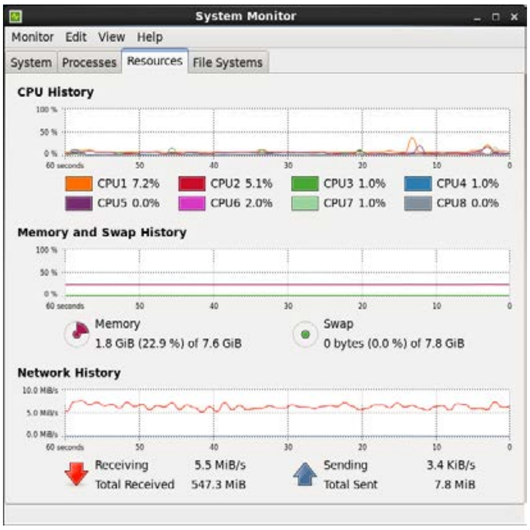

System Monitor
Monitor Edit View Help
System Processes Resources File Systems
CPU History
100 %
50 %
0 %
60 seconds 50 40 30 20 10 0
CPU1 7.2% CPU2 5.1% CPU3 1.0% CPU4 1.0%
CPU5 0.0% CPU6 2.0% CPU7 1.0% CPU8 0.0%
Memory and Swap History
100 %
50 %
0 %
60 seconds 50 40 30 20 10 0
Memory
1.8 GiB (22.9 %) of 7.6 GiB
Swap
0 bytes (0.0 %) of 7.8 GiB
Network History
10.0 MiB/s
5.0 MiB/s
0.0 MiB/s
60 seconds 50 40 30 20 10 0
Receiving 5.5 MiB/s
Total Received 547.3 MiB
Sending 3.4 KiB/s
Total Sent 7.8 MiB

Fig. 3. System parameters- Unicorn attack phase.

4.1 Data Preparation

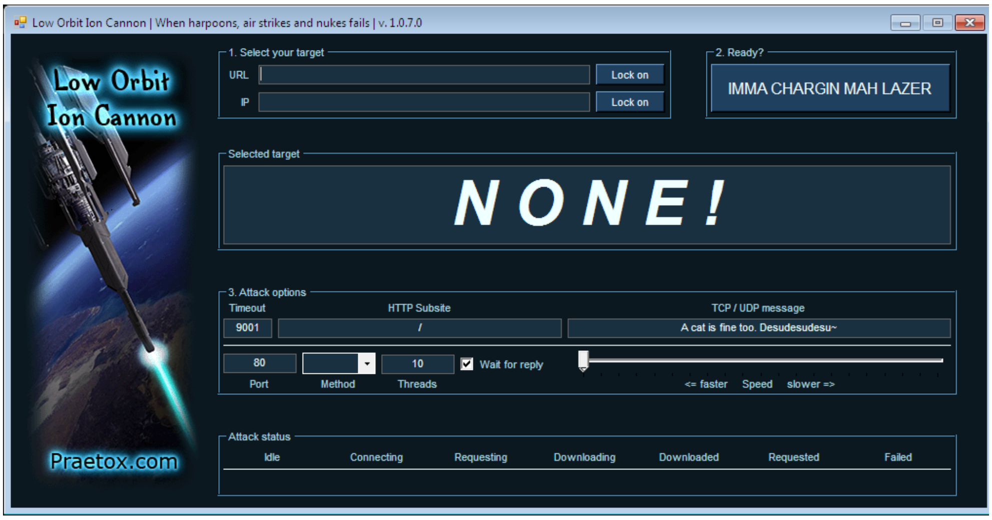
A total of 47 attributes are taken. For the initial no attacks case, a cloud application is run and the cloud system's different parameters in the fields of CPU, network, memory and storage are noted. These are the fundamental computing resources of a system; hence their analysis helps to understand the cloud system performance. When the system is subject to attack, there is a fluctuation in these values, which helps to identify the occurrence and onset of an attack. After features selection some redundant and irrelevant attributes are removed. The final list of attributes used for classification is shown in **Table 1**.

Table 1. List of different features and their description.

Feature	Description
CPU SIQ	Processes servicing soft interrupt requests.
CPU sys	Processes executing in kernel mode
CPU Usr	Processes executing in user mode
CPU wait	Processes waiting in queue
Disk read	Disk read in progress
Disk write	Disk write in progress
Frg	Socket IP fragments
IO read	IO read requests
IO write	IO write requests
Load 1m	Load statistics for 1 minute
Load 5m	Load statistics for 5 minutes
Load 15m	Load statistics for 15 minutes
Mem Used	Memory Used
Mem buff	Memory buffered
Mem free	Memory free
Network recv	Network received bytes
Process blk	Process Uninterruptable
Process new	Process New
Process run	Process Runnable
SDButil	Storage Disk B utilization
Sockets raw	Raw Sockets
System CSW	System Context Switches
TCP Clo	TCP Close
TCP syn	TCP Syn
TCP time	TCP Time wait
VM majpf	VM maj page faults
VM minpf	VM minj page faults

4.2 Tool Description

Low Orbit Ion Cannon (LOIC) is a denial of service and open source stress testing tool written in C#. It is used by many researches to perform DoS attacks because it is a GUI based tool and hence is easy to operate. It can send TCP, UDP and HTTP packets to the target machine.


Low Orbit Ion Cannon | When harpoons, air strikes and nukes fails | v. 1.0.7.0

1. Select your target

URL:

IP:

2. Ready?

Selected target: NONE!

3. Attack options

Timeout: 9001

HTTP Subsite: /

TCP / UDP message: A cat is fine too. Desudesudesu~

Port: 80

Method:

Threads: 10

☑ Wait for reply

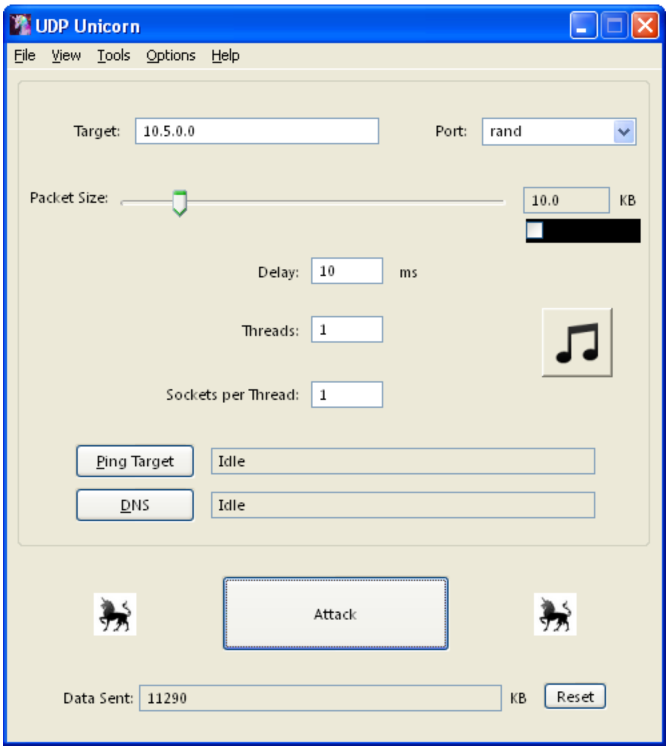
<= faster Speed slower =>

Attack status

Idle Connecting Requesting Downloading Downloaded Requested Failed

Fig. 4. A screen shot of the LOIC tool

Unicorn is a Dos testing tool based on http request written in C. It works in such a way that continuous http requests are sent to the server, so that its bandwidth gets exhausted and the genuine users are no longer able to access the resources they are entitled to.


UDP Unicorn

Target: 10.5.0.0

Port: rand

Packet Size: 10.0 KB

Delay: 10 ms

Threads: 1

Sockets per Thread: 1

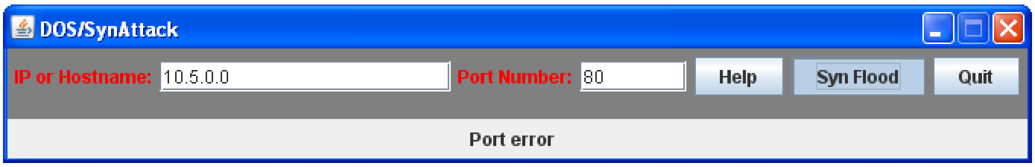
Ping Target: Idle

DNS: Idle

Data Sent: 11290 KB

Fig. 5. A screen shot of the Unicorn tool.

Similarly Pingflood, SynGUI and Pyloris are open source DoS tools that can be used to study attacks. They have many flexible options to increase the network payload and also the type of network packets to send like UDP, HTTP or TCP, the port to which this traffic has to be directed to, the number of packets to be send, the length of the packets, etc.,



IP or Hostname: 10.5.0.0
Port Number: 80

Port error

Fig. 6. A screen shot of the SynGUI tool.

5. Results and Discussions

The dataset is divided into 10 folds and 9 folds were used for training. As we are dealing with a multiclass classifier problem, the analysis of the performance parameters is complex. There are 6 classes of instances in the dataset, namely, 1) No attack, 2) SynGUI attack, 3) Unicorn attack, 4) Pyloris attack, 5) LOIC attack, 6) Ping Flood attack.

In the feature selection phase, we compare the accuracy achieved with and without using manual feature selection in **Fig. 7,** against Decision Tree classifier and the intelligent rule based classifier. It can be observed that the accuracy is increased after applying feature selection.

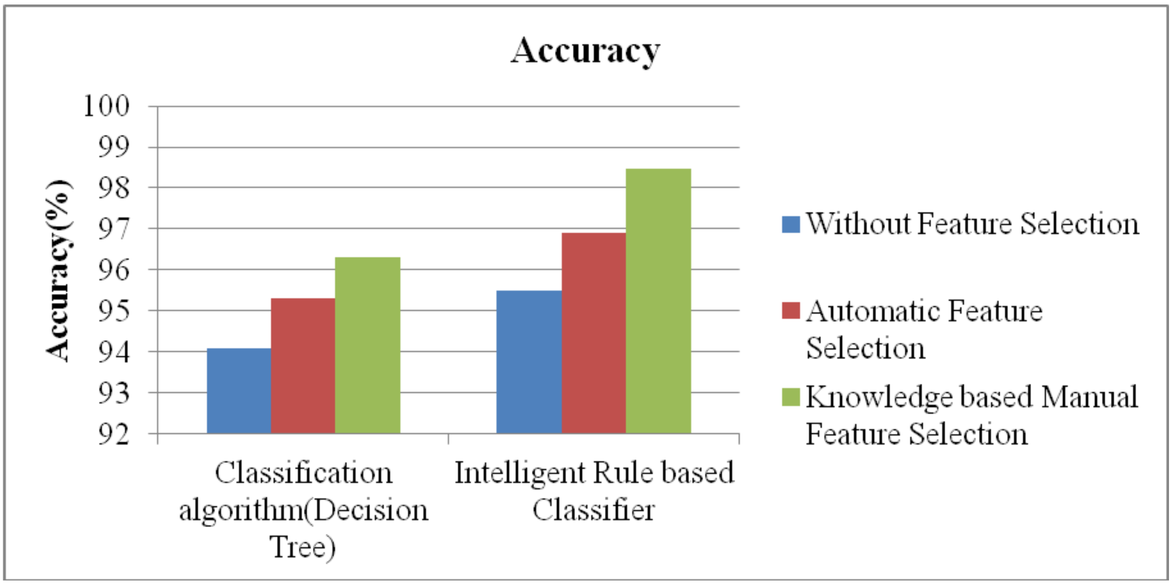


Fig. 7. Accuracy with and without feature selection.

For ease in comparing the results and understanding them better, the total dataset of 10,000 instances is being split into different experimental sets. Experiment 1 consists of 100 instances in each of the types mentioned above. Likewise, experiment 2 consists of 200 instances in each type, experiment 3 has 500 instances each, experiment 4 has 1000 of each type and experiment 5 is the full dataset with 1000 instances in each attack type and 5000 instances in the no attack category.

Table 2. Confusion matrix in classification.

		Results of Classifier Prediction	
		Positive	**Negative**
Actual Values	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

The various parameters that have been used for analysis purpose are:

$$\text{Precision} = \text{True Positive} / (\text{True Positive} + \text{False Positive}) \qquad (12)$$

$$\text{Recall} = \text{True Positive} / (\text{True Positive} + \text{False Negative}) \qquad (13)$$

$$\text{F-measure} = 2*(\text{Precision}*\text{Recall}) / (\text{Precision} + \text{Recall}) \qquad (14)$$

$$\text{False Positive Rate} = \text{False Positive} / (\text{True Positive} + \text{False Positive}) \qquad (15)$$

Table 3. Precision, Recall and F-measure values for the various experiments carried out.

Parameter	**No Attack**	**SynGUI**	**Unicorn**	**Pyloris**	**LOIC**	**Ping Flood**
Experiment 1						
Precision	0.78	0.767	0.92	0.824	0.82	0.85
Recall	0.86	0.79	0.84	0.75	0.81	0.9
F-Measure	0.82	0.78	0.88	0.79	0.81	0.87
Experiment 2						
Precision	0.87	0.92	0.9	0.91	0.925	0.94
Recall	0.95	0.905	0.825	0.92	0.925	0.935
F-Measure	0.91	0.91	0.86	0.91	0.925	0.94
Experiment 3						
Precision	0.9	0.95	0.96	0.95	0.96	0.95
Recall	0.964	0.95	0.938	0.94	0.922	0.948
F-Measure	0.93	0.95	0.95	0.94	0.93	0.95
Experiment 4						
Precision	0.95	0.98	0.98	0.98	0.99	0.986
Recall	0.984	0.986	0.979	0.969	0.963	0.987
F-Measure	0.96	0.984	0.97	0.97	0.97	0.98
Experiment 5						
Precision	0.98	0.97	0.98	0.97	0.979	0.98
Recall	0.998	0.984	0.955	0.955	0.979	0.983
F-Measure	0.99	0.97	0.97	0.96	0.979	0.98

The precision, recall and f-measure values computed for the different experiments have been tabulated in **Table 3**. The values have been tabulated for individual types of attacks. These are a gradual increase in the values as the size of the dataset increases. Precision is useful in expressing the exactness of the classifier and recall depicts its completeness. And they range from 0 to 1. The closer they are to 1, the better. From our experimental results we can see that the No Attack class and the Ping Flood attack class have higher precision and recall. F-measure is a measure used to tell about the test accuracy. Both precision and recall are taken into consideration while calculating F-measure. It ranges from 0 to 1 and the nearer it is to 1, the better the teat accuracy of the system. In our experiments we can observe that the value is more for the datasets which have more instances.

Table 4. Aggregation of the Precision, Recall and F-measure values for all the experiments in shown in Table 3.

Split in the data	Precision	Recall	F-Measure
Experiment 1	0.83	0.825	0.83
Experiment 2	0.91	0.91	0.91
Experiment 3	0.94	0.94	0.94
Experiment 4	0.97	0.978	0.97
Experiment 5	0.98	0.98	0.97

The classifier accuracy gets increased when the dataset size is increased, this happens because the classifier algorithm has more examples to see and learn and perfect its facts. This can be seen in Table 4.

Table 5. Comparing intelligent rule based classifier results with Decision Tree.

Split in the data	Accuracy	
	With Classification (Decision Tree)	Using Expert Rules
Experiment 1	80.9%	82.5%
Experiment 2	92.3%	91%
Experiment 3	93.5%	94.36%
Experiment 4	95.8%	97.8%
Experiment 5	96.3%	98.46%

The accuracy of Decision Tree for the different experiment batches is compared with our intelligent rule based classifier in Table 5, and it can be inferred that thought the accuracy of Decision Tree increases as the dataset size increases; the intelligent rule based classifier has higher accuracy throughout.

Table 6. Comparison with existing works.

Classification Algorithm	Accuracy
Naïve Bayes	68.53%
Multilayer Perceptron	94.33%
Support Vector Machine	96.27%
Decision Tree	74.71%
PART	73.44%
Intelligent Rule based Classifier	98.46%

The accuracy of intelligent rule based classifier is compared with the existing works [15] in Table 6. It can be clearly seen that by formulating our own rule based classifier, we have achieved greater accuracy.

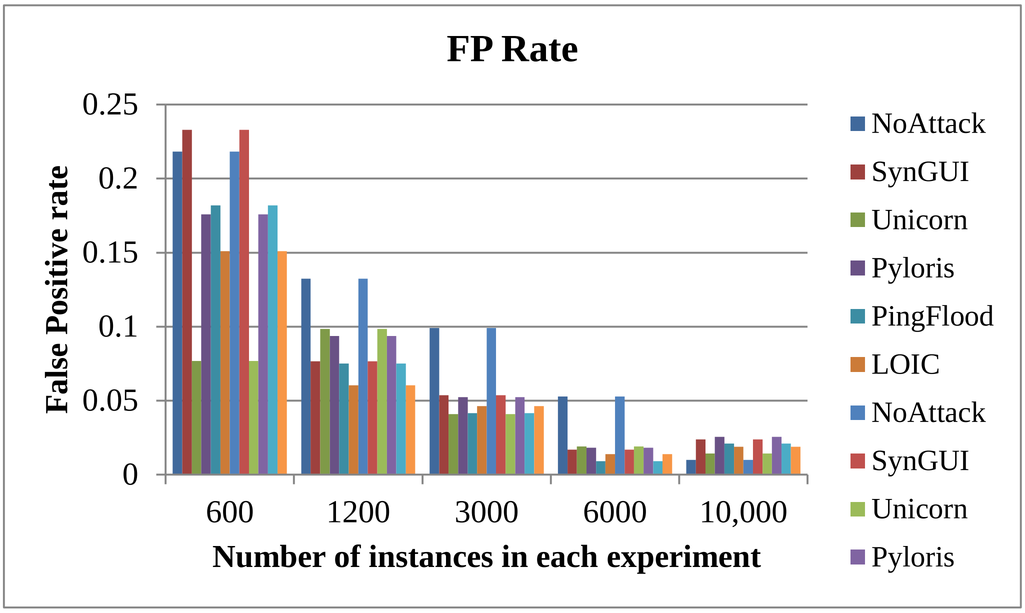


Fig. 8. Graph comparing the FP rate across different classes and experiments.

False positive is a state where an instance has been falsely predicted as positive. False positive rate is an important parameter in crucial applications like spamming and phishing web site detection, intrusion detection and bio medical screening application. Lower the false positive rate, the better. From the results of our experiment, we can see in **Fig. 8** that Unicorn class instances are, throughout, displaying lesser false positive rate. We do not want the FP rate to be high for the No Attack class as we do not want any attack to be predicted as a no attack class. We can observe from the results that in the final batch of 10,000 instances, the FP-rate of No attack class has been considerably reduced.

6. Conclusion

In the present internet scenario, cloud is expanding rapidly and the need of the hour is to tap into its flexibility and dynamic resource allocation features. But the disadvantages of the existing internet scenario, viz the different attacks on privacy and secrecy of customer data should not be allowed to surge in cloud environment as well. So researchers have to build novel and intelligent intrusion detection systems to detect and terminated these attack scenarios. In this paper, we have successfully detected DoS attacks using an intelligent rule based classifier and also the applied knowledge based feature selection. We conclude that domain specific intelligent system will be able to counter the problems in a more efficient manner; as compared to the existing classifier algorithms when there is ample expert advice available. The system is more suitable for enhancing security of web applications such as e-commerce, tele-medicine and e-learning which are deployed in cloud for enhancing the reliability and availability. As a future extension of this work, we intend to add a fuzzy logic component to this system that can make the detection of attacks faster and more efficient.

References

[1] Nguyen Dinh Han, Yonghwa Chung and Minho Jo, "Green data centers for cloud-assisted mobile ad hoc networks in 5G," *IEEE Networks*, vol. 29, no. 2, pp. 70-76, April, 2015. Article (CrossRef Link)

[2] P.Vijayakumar, S.Bose, A.Kannan and L.Jegatha Deborah, "Computation and Communication Efficient Key Distribution Protocol for Secure Multicast Communication," *KSII Transactions on Internet & Information Systems*, vol. 7, no. 4, pp. 878-894, April, 2013. Article (CrossRef Link)

[3] Seung Yeob Nam and Sirojiddin Djuraev, "Defending HTTP Web Servers against DDoS Attacks through Busy Period-based Attack Flow Detection," *KSII Transactions on Internet & Information Systems*, vol. 8, no. 7, pp. 2512-2531, July, 2014. Article (CrossRef Link)

[4] Zhe Li, Weiqing Sun and Lingfeng Wang, "A Neural Network Based Distributed Intrusion Detection System On Cloud Platform," in *Proc. of 2nd IEEE Conf. on Cloud Computing and Intelligence Systems*, pp. 75-79, October 30-November 1, 2012. Article (CrossRef Link)

[5] Ahmed Patel, Mona Taghavi, Kaveh Bakhtiyari and Joaquim Celestino Júnior, "An intrusion detection and prevention system in cloud computing: A systematic review," *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 25-41, January, 2013. Article (CrossRef Link)

[6] Wayne A. Jansen, "Cloud Hooks: Security and Privacy Issues in Cloud Computing," in *Proc. of 44th Hawaii International Conf. on System Sciences*, January 1, 2011. Article (CrossRef Link)

[7] Meiko Jensen, Jorg Schwenk, Nils Gruschka and Luigi Lo Iacono, "On Technical Security Issues in Cloud Computing," in *Proc. of 2nd IEEE International Conf. on Cloud Computing*, pp. 109-116, September 21-25, 2009. Article (CrossRef Link)

[8] Thomas Ristenpart, Eran Tromer, Hovav Shacham and Stefan Savage, "Hey, You, Get Off of My Cloud: Exploring Information Leakage in Third-Party Compute Clouds," in *Proc. of 16th ACM Conf. on Computer and Communications Security*, pp. 199-212, November 9-13, 2009. Article (CrossRef Link)

[9] Shui Yu, Yonghong Tian, Song Guo, and Dapeng Oliver Wu, "Can We Beat DDoS Attacks in Clouds?," *IEEE Transactions On Parallel And Distributed Systems*, vol. 25, no. 9, pp. 2245-2254, September, 2014. Article (CrossRef Link)

[10] Qiao Yan and Yu .F, "Distributed denial of service attacks in software-defined networking with cloud computing," *IEEE Communications*, vol. 53, no. 4, pp. 52-59, April, 2015. Article (CrossRef Link)

[11] Girma Anteneh, Garuba Moses, Li Jiang and Liu Chunmei, "Analysis of DDoS Attacks and an Introduction of a Hybrid Statistical Model to Detect DDoS Attacks on Cloud Computing Environment," in *Proc. of 12th International Conf. on Information Technology- New Generations*, pp. 212-217, April 13-15, 2015. Article (CrossRef Link)

[12] Ferriman.B, Hamed.T and Mahmoud.Q.H, "Storming the cloud: A look at denial of service in the Google App Engine," in *Proc. of International Conf. on Computing, Networking and Communications*, pp. 363-368, February 16-19, 2015. Article (CrossRef Link)

[13] Bing Wang, Yao Zheng, Wenjing Lou and Hou, Y.T., "DDoS attack protection in the era of cloud computing and Software-Defined Networking," in *Proc. of 22nd IEEE International Conf. on Network Protocols*, pp. 624-629, October 21-24, 2014. Article (CrossRef Link)

[14] Ahmad Karim, Syed Adeel Ali Shah, Rosli Bin Salleh, Muhammad Arif, Rafidah Md Noor and Shahaboddin Shamshirband, "Mobile Botnet Attacks – an Emerging Threat: Classification, Review and Open Issues," *KSII Transactions on Internet & Information Systems*, vol. 9, no. 4, pp. 1471-1492, April, 2015. Article (CrossRef Link)

[15] Oktay.U and Sahingoz O.K, "Attack Types and Intrusion Detection Systems in Cloud Computing," in *Proc. of 6th International Conf. on Information Security & Cryptology*, pp. 71-76, May 23-24, 2013.

[16] Hansung Lee, Daesung Moon, Ikkyun Kim, Hoseok Jung and Daihee Park, "Anomaly Intrusion Detection Based on Hyper-ellipsoid in the Kernel Feature Space," *KSII Transactions on Internet & Information Systems*, vol. 9, no. 3, pp. 1173-1192, March, 2015. Article (CrossRef Link)

[17] Md Tanzim Khorshed, A B M Shawkat Ali and Saleh A. Wasimi, "Monitoring Insiders Activities

in Cloud," in *Proc. of 10th International Conf. on Trust, Security and Privacy in Computing and Communications*, pp. 757-764, November 16-18, 2011. Article (CrossRef Link)

[18] Ashley Chonka, Yang Xiang, Wanlei Zhou and Alessio Bonti, "Cloud security defence to protect cloud computing against HTTP-DoS and XML-DoS attacks," *Journal of Network and Computer Application*, vol. 34, no. 4, pp. 1097-1107, June, 2011. Article (CrossRef Link)

[19] Junaid Arshad, Paul Townend and Jie Xu, "A novel intrusion severity analysis approach for Clouds," *Future Generation Computer Systems*, vol. 29, no. 1, pp. 416-428, January, 2013. Article (CrossRef Link)

[20] Chirag Modi, Dhiren Patel, Bhavesh Borisaniya, Hiren Patel, Avi Patel and Muttukrishnan Rajarajan, "A survey of intrusion detection techniques in Cloud," *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 42-57, January, 2013. Article (CrossRef Link)

[21] Kapil Kumar Gupta, Baikunth Nath and Ramamohanarao Kotagiri, "Layered Approach Using Conditional Random Fields for Intrusion Detection," *IEEE Transactions on Dependable and Secure Computing*, vol. 7, no. 1, pp. 35-49, March, 2010. Article (CrossRef Link)

[22] Sannasi Ganapathy, Kanagasabai Kulothungan, Sannasy Muthurajkumar, Muthuswamy Vijayalakshmi, Yogesh Palanichamy and Arputharaj Kannan, "Intelligent feature selection and classification techniques for intrusion detection in networks: a survey," *EURASIP Journal on Wireless Communications and Networking*, no. 1, pp. 1-16, November, 2013. Article (CrossRef Link)

[23] Siva S Sivatha Sindhu, S Geetha and A Kannan, "Decision tree based light weight intrusion detection using a wrapper approach," *Expert Systems with applications*, vol. 39, no. 1, pp. 129-141, January, 2012. Article (CrossRef Link)

[24] Jaeik Cho, Taeshik Shon, Ken Choi and Jongsub Moon, "Dynamic learning model update of hybrid-classifiers for intrusion detection," *The Journal of Supercomputing*, vol. 64, no. 2, pp. 522-526, May, 2013. Article (CrossRef Link)

[25] Siva S Sivatha Sindhu, S Geetha and A Kannan, "Evolving optimized decision rules for intrusion detection using particle swarm paradigm," *International Journal of Systems Science*, vol. 43, no. 12, pp. 2334-2350, December, 2012. Article (CrossRef Link)

[26] Md Tanzim Khorshed, A B M Shawkat Ali and Saleh A. Wasimi, "Classifying different denial-of-service attacks in cloud computing using rule-based learning," *Security and Communication Networks*, vol. 5, no. 11, pp. 1235-1247, November, 2012. Article (CrossRef Link)

[27] Md. Tanzim Khorshed, A.B.M. Shawkat Ali and Saleh A. Wasimi, "A survey on gaps, threat remediation challenges and some thoughts for proactive attack detection in cloud computing," *Future Generation Computer Systems*, vol. 28, no. 6, pp. 823-851, June, 2012. Article (CrossRef Link)

[28] Quinlan J.R, "Induction of Decision Trees," *Machine Learning*, vol. 1, no. 1, pp. 81-106, March, 1986. Article (CrossRef Link)

Pradeepthi.K.V is a Ph.D student in the Department of Information Science and Technology, College of Engineering, Guindy, Anna University. Her research interests include Cloud Computing, Data Mining, Machine Learning and Web Security.

Kannan.A is the Head of the Department and Professor in the Department of Information Science and Technology, College of Engineering, Guindy, Anna University. His areas of interest include Database Management Systems, Network Security, Software Engineering, Machine Learning and Cloud Computing. He has published more than 200 papers in International journals and conference proceedings. He has also authored one book on database management systems and one book on computer programming. He has more than 31 years of experience in teaching and research.