Intelligent Intrusion Detection System for Private Cloud Environment

Muthukumar B.[1(✉)] and Praveen Kumar Rajendran[2]

[1] Faculty of Computing, Sathyabama University, Chennai, India
anbmuthusba@gmail.com
[2] Programmer Analyst Trainee,
Cognizant Technology Solutions, Chennai, India
praveenkumar558@gmail.com

Abstract. From the day cloud computing got its popularity, security and performance is the two important issues faced by the cloud service providers and the clients. Since cloud computing is a virtual pool of resources provided in an open environment (Internet), identifying intrusion of unauthorized users is one of the greatest challenges of the cloud service providers and cloud users. The artificial intelligence technique has been proposed in this paper in order to identify the intrusion of unauthorized user in a cloud environment. Application or research on cloud computing is always primarily focused upon any one of the issues. In our paper, the proposed algorithm satisfies the security aspects of cloud computing and the performance testing of the implementation satisfies the performance issues of cloud computing.

Keywords: Cloud computing · Intrusion · Intrusion detection system · Intelligence intrusion detection

1 Introduction

Cloud computing has made tremendous changes in the functioning and working in Information Technology sector As a result in exponential growth of data, the organization started to invest more on building their infrastructure which increased the capital expenditure of an organization. Cloud Computing has also changed the way in which business and personal data are being stored and accessed using computer, which has led to many kind of security issues [11]. Providing security for the data that has been stored in the cloud is one of the important responsibilities of the service provider. Although the infrastructure of the cloud is much more reliable, it faces lot of internal and external threats [12]. Hacking, Intrusion are the two major threat and security issues in cloud computing [3]. Activities of hacking can be easily identified on a network. Identification of intrusion in a network is quite tedious. An Intrusion Detection system that can identify the intrusion in an efficient manner and work as per the nature of cloud computing will give a solution for the security issue of cloud computing. An Intelligence Intrusion Detection system has been proposed in this paper, which would be another step in research on security aspects of cloud computing.

J.H. Abawajy et al. (Eds): SSCC 2015, CCIS 536, pp. 54–65, 2015.
DOI: 10.1007/978-3-319-22915-7_6

2 Cloud Computing

In cloud computing All the service are hosted via the Internet by service provider and used via Internet virtually, which leads to Internet intrusion [2]. Via cloud computing, the basic requirements of a customer are provided as a service. Software, Infrastructure, Platform are provided as a service by the service providers. In short, anything is provided as a service to the clients [15]. These Cloud services are provided in various manners such as "Public Cloud", "Private Cloud", and "Community Cloud". The main characteristic of the cloud service is "Pay as you go manner". It means the client has to pay only for the service which has been utilized.

Many research scholars and scientist have defined cloud computing at various occasions. Buyya et al. [1] has defined cloud computing as follows "Cloud is a parallel and distributed computing system consisting of a collection of inter-connected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resources based on service-level agreements (SLA) established through negotiation between the service provider and consumers." From Buyya et al. [1] it can be inferred that cloud computing has the base of parallel computing, distributed computing, virtualization. Among these concepts, virtualization plays a major role in cloud computing. The major challenge before the research scholars is to provide a security for the transactions made and security for the data that is being stored [3, 7].

3 Intrusion and Intrusion Detection Systems

According to the Sundaram et al. [6], the term intrusion can be defined as "… the act of detecting actions that attempt to compromise the confidentiality, integrity or availability of a resource". The term Intrusion Detection is a field of research and development, which generally deals with intrusion and abnormal activity in a computer or in a network [4]. Intrusion, can be generally classified into two major categories such as Misuse Intrusion Detection and Anomaly Intrusion Detection. Intrusion Detection System can be classified as Network Based Intrusion Detection System and Host Based Intrusion Detection System.

Misuse Intrusion Detection is generally a signature based or rule based intrusion detection method. In misuse intrusion detection, the intrusion is identified when something happens apart from the set of rules that has been fixed by the administrator. The major drawback with this type of intrusion detection is, the rules have to be updated in a constant manner [7]. Iterative and Genetic are the two major types of Misuse intrusion detection method. Iterative detect intrusion in a continuous manner, where genetic method detects using previous history [13]. In anomaly intrusion, the intrusion is identified using the previous history of intrusion. Whenever an intrusion has been detected, the record of the intrusion will be stored in the database. If the same pattern of activity occurs in the future, using the stored pattern intrusion will be determined [7]. Static anomaly intrusion detection and dynamic anomaly intrusion are the two major types of anomaly intrusion detection method [14].

In general Intrusion detection system is either hardware or an application. If this hardware component of the application would be in the common place of a network

and monitor the entire network's activity, it is said to be Network based Intrusion Detection System. David J. Weller-Fahy et al. [16] has defined network based intrusion detection system as an automated system that detects the intrusion in a network. The major drawback with the Network based Intrusion Detection System is, these systems cannot monitor the activities of each and every node that are present in the network In order to monitor the activities of each and every node in a network, Host Based Intrusion Detection System has been used. Host based Intrusion detection will not monitor the activities of the other host in the system or monitor the network [17].

3.1 Research Motivation

The main goal of research in Intrusion Detection System is to build an efficient intrusion detection system, which can detect any type of intrusion within the host as well as in the network. Patel et al. [7] describe, "Elasticity, Reliability, Agility and Adaptability, Availability", are some of the basic characteristics of cloud computing. Among the basic characteristics of cloud computing, the major challenge for implementing an intrusion detection system would be elasticity and adaptability. The intrusion detection system has to adopt it as per the nature of the cloud and also should work in an efficient manner even if there is any change in the nature of the cloud. Building an efficient intrusion detection system for an elastic environment is another major challenge for the researchers and developers. This can be possible, by embedding intelligent technique (Based upon Artificial Intelligence) within the intrusion detection system. In order to build such intelligent intrusion detection system with high security and performance, various research works have been studied.

3.2 Related Works

During our research work on intrusion detection and intrusion detection system, a detailed study on previous work was done and proposed a Hybrid Intrusion detection mode [5]. Our previous research work narrate that, intrusion detection system should be dynamic, self adaptive, scalable and efficient in nature.

Kleber Vieira et al. [8] have proposed a Hybrid Intrusion Detection System for Cloud computing and Grid computing environment. The proposed Intrusion detection system can detect only one type of intrusion i.e., either anomaly intrusion or misuse intrusion. The architecture and working of cloud computing and grid environment are completely different. Proposing a common intrusion detection system for two different concepts is contradictory. From our analysis we could infer that the proposed intrusion detection system is not efficient in terms of detecting new type of intrusion and hence it does not update the system. This system is more suitable for grid environment, than cloud based environment.

Tupakula et al. [9] have proposed a Virtual machine based Hybrid Intrusion Detection System for Infrastructure as a Service. From our analysis [19] we inferred that, this system cannot handle real time environment and implementing the intrusion detection system for other cloud services has not been defined. The system which has

been narrated is quite complex and implementation steps (algorithm) for implementing the system has not been defined. The scope of implementing the system in a real time cloud based environment has not been discussed by the authors. And this system does not satisfy the characteristics of Hybrid Intrusion Detection system, which has been proposed in our earlier research work [5].

Kholidy et al. [10] have proposed a framework for Intrusion Detection in Cloud Systems. This framework isn't efficient as per our analysis and the same has been validated by Patel et al. [7]. As per our analysis, this system does not detect intrusion in fast and efficient manner.

As per our analysis made on earlier research works the following inferences were made.

1. Existing works are not Dynamic, Self adaptive, Efficient and Scalable in nature.
2. Proper algorithm for implementing the intrusion detection system has not been defined.
3. Existing intrusion detection systems are complex and difficult to implement.
4. Less efficient in detecting different types intrusion.
5. Performance has not been considered.

The following Intelligent Intrusion Detection Algorithm will overcome the drawbacks of the existing systems and also satisfy the characteristics of Intrusion Detection System proposed in our earlier research work [5].

4 Intelligent Intrusion Detection

The main goal of the Intelligent Intrusion Detection System is to detect the intrusion in an efficient manner with the help of previous history of intrusion and by updating the intrusion detection database in a constant manner. The purpose of introducing intelligence, technique is to detect intrusion in an efficient manner by predicting the intrusion using the training given to the system. The intelligence intrusion detection system has been proposed by combining hardware and an application to detect the intrusion. The 3 major phases of the proposed intrusion detection are.

1. Training the intrusion detection system.
2. Testing the intrusion detection system.
3. Implementation and updating intrusion detection system.

4.1 Muthu-Praveen Algorithm of Intelligent Intrusion

The above proposed Intelligent Intrusion Detection System can be implemented using the following algorithms. Each phase which has been discussed in Sect. 4 has been written as an algorithm.

Algorithm 1: Training the Intrusion Detection System

In the training phase, the hardware component and the application will be trained with sample intrusion data. If any such trace is found during the implementation phase, the

application and the hardware will detect the intrusion based upon the training given. The sample intrusion data contains abnormal port number and protocol used by the end user, abnormal path through which the request has travelled.

```
1.      Start
2.  {
3.  Initialize the hardware component;
4.  Initialize the software component;
5.  {
6.  Fetch the sample intrusion data into the hardware component;
7.  {
8.  Store the data in the database of the hardware component;
9.  }
10. Fetch the sample intrusion data into the application;
11. {
12. Store the data in the database of the hardware component;
13. }
14. }
15. End
```

Algorithm 2: Testing the intrusion detection system

After training the intrusion detection system with the sample data, the system is tested to check whether the training has been done successfully. Testing phase of intrusion detection system can be implemented using the following Algorithm 2. In this phase, the system will be fetched with the similar kind of kind of data which has been used during the training phase. System should identify the trace of intrusion perfectly, if not the system will be trained once again.

```
1.  Start
2.  {
3.  Initialize the testing phase;
4.  Fetch the sample intrusion data to the hardware component and software
    component;
5.  If the system is able to detect the intrusion
6.  {
7.  Training is good and the testing is successful;
8.  }
9.  Else
10. {
11. Training is not good;
12. Testing phase is not successful;
13. System is recommended for training once again;
14. }
```

Algorithm 3: Implementing and Updating of Intelligent Intrusion Detection System

Once the training and testing is successful, the system will be exposed to the real time scenario. If the system identifies the similar kind of intrusion trace, the system will intimate the cloud admin and the users. If any new trace has been found, the trace will be stored in the database and will be used in the future training process. The following Algorithm 3 can update itself, without any human intervention (Intelligent mechanism).

```
1. Start
2. {
3. For all the incoming request;
4. Fetch the intrusion detection parameters from the request;
5. For all the parameters
6. {
7. Check with the values stored in the database
8. }
9. If there is any deviation in the parameter
10. {
11. Intimate the cloud administrator about the intrusions;
12. }
13. If the deviation found is proved to be an intrusion trace
14. {
15. Pass the new intrusion trace to the data base;
16. Update the database;
17. }
18. If the deviation found is wrong
19. {
20. Store the traces in the database;
21. }
22. For all the proved intrusion traces
23. {
24. Use the trace that is stored in the database to train the system;
25. Intrusion alert will be made for such trace in the future;
26. }
27. For all the deviation which is found wrong
28. {
29. Use the trace that is stored in the database to train the system;
30. The system will not intimate about intrusion for such trace;
31. }
32. }
33. End
```

4.2 Implementation

In the above proposed algorithms software component is implemented using the.net as front end and SQL server as the back end. In order to implement the proposed intrusion detection algorithm in an open source environment such as open stack, it can be implemented using the open source languages PHP, Perl or Python with the same SQL server as the backend. Since the algorithm has been written in such a way that it will predict the deviations perfectly, the system will work more efficiently than earlier proposed systems. Index Page (Application Home Page), User Login Page, User Home Page, Admin Page has been created and the above proposed intrusion detection algorithm has been implemented in the above said pages.

4.3 Performance Evaluation

The performance of the implemented algorithm is measured using an open source performance testing tool, JMeter. In JMeter, the performance of the application is measured in terms of response time. In order to prove that application is functionally good, error criteria from the JMeter have been considered. Figure 1 shows the snapshot of Performance evaluation process (Aggregate Report), which is formed as Table 1.

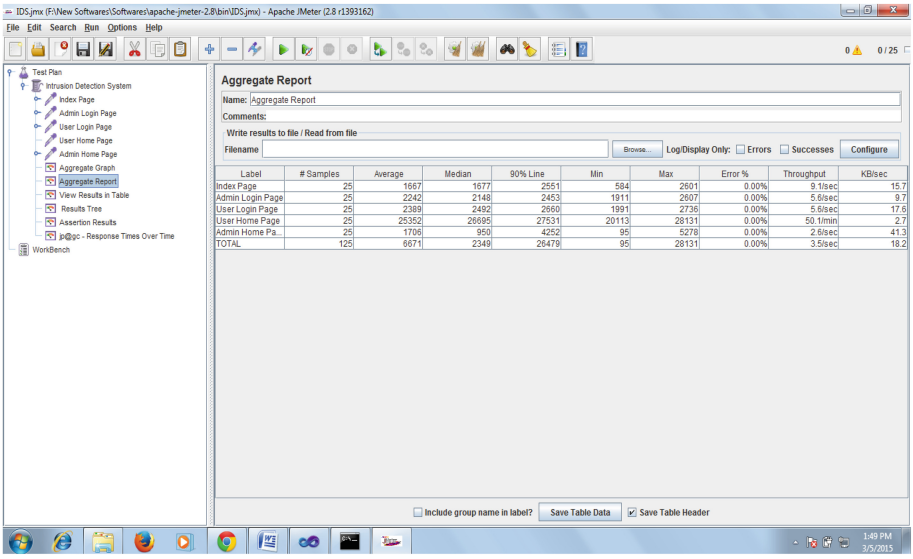

Fig. 1. Snapshot of performance analysis

4.4 Result of Performance Evaluation

The Fig. 2 and a Table 1 have been obtained to demonstrate the result of performance evaluation. In the Table 1, the term “label” indicates the pages which have been tested. “Sample” indicates the number of times (Users) the page has been tested. “Response

time" will give the average time taken by each page. "Error %" indicates, whether the page has any functional error. Other parameters are included in the paper to show the originality, which are not considered in this paper. Future analysis and research can be carried out using those parameters. For readability purpose, the values derived from the tool are described with the respective units in the following Table 1.

Table 1. Performance evaluation for 5 users

Label	Samples (Count)	Average response Time (ms)	Median (ms)	90 % line (ms)	Min (ms)	Max (ms)	Error (%)	Throughput (Count/ms)	Data transfer (KB/ms)
Index Page	25	1667	1677	2551	584	2601	0	9.067827	15.74472
Admin Login Page	25	2242	2148	2453	1911	2607	0	5.581603	9.691494
User Login Page	25	2389	2492	2660	1991	2736	0	5.55679	17.64172
User Home Page	25	25352	26695	27531	20113	28131	0	0.835366	2.701071
Admin Home Page	25	1706	950	4252	95	5278	0	2.554409	41.26468
Total	125	6671	2349	26479	95	28131	0	3.494939	18.19826

4.5 Inference of Performance Evaluation

In the Fig. 2, Red line indicates Index page, Blue line indicates Admin Login Page, Pink color indicates User login page, Light blue color indicates Admin Home page and

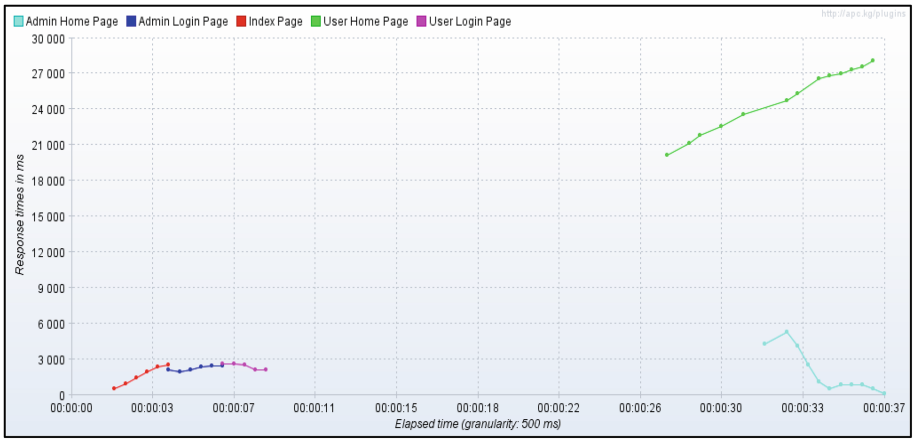

Fig. 2. Response time graph (Color figure online)

Green color indicates User home page. In the Fig. 2, X axis indicates the elapsed time and Y axis indicates the response time of each page that has been tested. The following are the inferences, which can be concluded from the Table 1 and Fig. 2.

1. The total Error percentage of the application is Zero percentage (0 %). Zero percentage 0 % of Error, shows that the application is functionally good (without any error, the application has passed in all the test cases).
2. The average response time of the application with 25 users is 6.67 s, which is less than 7 s. This indicates that 25 users can access the application within 7 s, from which we can infer that application holds good in terms of performance.
3. From the Table 1, we can infer that the response time range of 4 pages of the application is less than 2.6 s. From this result, we can conclude that 90 % of performance of the application holds good.
4. Range of response time for User Home page and Admin Home Page depends upon the content that has been created in that particular page. In our research, a sample Home page and admin page has been created and the testing has been carried out.
5. In the Fig. 2, the ranges of the response time between highest value and the lowest value are plotted. Highest and the lowest range of 4 pages (Index, Admin Login, User Login, Admin Home Page) are less than 6 s, and the range of User Home page is higher due to the content of the home page.

4.6 Statistical Analysis

In order to prove the correctness of the proposed algorithm, One-way Analysis of Variance test has been carried out using Minitab 17 (Trial Version). The main purpose of performing ANOVA test is to find the mean difference within the group and different group [19]. Here the mean difference within the group has been considered. One-way ANOVA test has been carried out between the mean values of Response time across the Sample value. Figure 3 and Table 2 is the result obtained from the One-way ANOVA test.

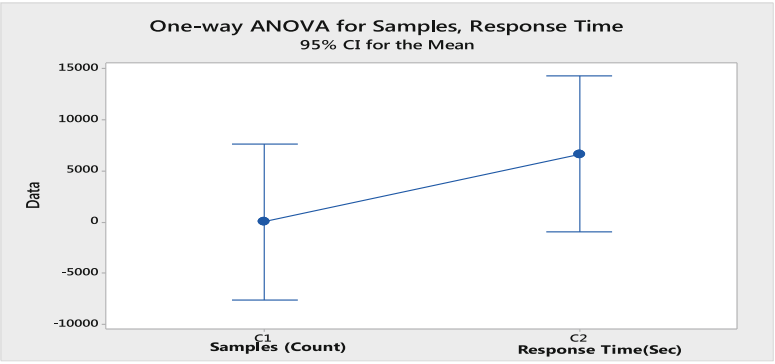

Fig. 3. One-way analysis of variance

In the Fig. 3, Samples are taken as the 1st factor and Response time is taken as the second factor and ANOVA test has been carried out. Table 2 has been derived as an output of one-way ANOVA process from Minitab 17 (trail version). Here F indicates the factor value, R indicates the Response value and CI indicates the confidence interval. For our research purpose, these parameters are considered. In our research, response is "Response time" and the factor that influences response time is "Samples". In the Table 2, C1 indicates the Response time and C2 indicates the sample.

Table 2. Result of one-way anova

Method

Null hypothesis All means are equal
Alternative hypothesis At least one mean is different
Significance level $\alpha = 0.05$

Equal variances were assumed for the analysis.

Factor Information

Factor	Levels	Values
Factor	2	C1, C2

Analysis of Variance

Source	DF	Adj SS	Adj MS	F-Value	P-Value
Factor	1	110429936	110429936	2.02	0.193
Error	8	436622567	54577821		
Total	9	547052503			

Model Summary

S	R-sq	R-sq(adj)	R-sq(pred)
7387.68	20.19%	10.21%	0.00%

Means

Factor	N	Mean	StDev	95% CI
C1	5	25.00	0.00	(-7593.74, 7643.74)
C2	5	6671	10448	(-948, 14290)

Pooled StDev = 7387.68

4.7 Discussion

In the earlier research work, only the security aspects have been considered. In the proposed algorithm both security and performance aspect has been considered. The performance of the proposed algorithm may vary based upon the nature of the cloud. If the proposed algorithm is deployed in the public cloud, the security parameters and

Table 3. Comparison of proposed algorithm

S.no	Research work	Comparison criteria	
		Security	Performance
1	Kleber Vieira et al. [8]	Good	Not considered
2	Tupakula et al. [9]	Good	Not considered
3	Kholidy et al. [10]	Good	Not considered
4	Proposed Algorithm	Good	Considered

the performance parameters would vary at a large extent. The result of performance testing gives an overall impression, that the implementation is much efficient in terms of time and space (Table 3).

5 Future Scopes and Conclusion

In the private cloud environment, the proposed algorithm was able to detect all of new types of intrusion (100 %). The framework of the algorithm can be expanded and implemented in other cloud deployment models. Implemented application and the entire algorithm can be deployed in highly secured private cloud such as cloud that is being built for defense purpose, educational purpose etc. Nader Sohrabi Safa et al. [18] al has proposed a method to identify the customers using Artificial intelligence method; research using the concept proposed by Nader Sohrabi Safe et al. [18] would make the research on Intrusion detection much more interesting and also will make a new dimension in network security.

References

1. Buyya, R., Yeo, C.S., Venugopal, S., Broberg, J., Brandic, I.: Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility. Future Gener. Comput. Syst. **25**(6), 599–616 (2009)
2. Zarrabi, A., Zarrabi, A.: Internet intrusion detection system service in a cloud. Int. J. Comput. Sci. **9**(1), 308 (2012)
3. Kandukuri, B.R., Paturi, R.V., Rakshit, A.: Cloud security issues. In: 2009 IEEE International Conference on Services Computing, Bangalore, 21–25 September 2009
4. Jabez, J., Muthukumar, B.: Intrusion detection system: time probability method and hyperbolic hopfield neural network. J. Theor. Appl. Inf. Technol. **67**(1), 65–77 (2014)
5. Rajendran, P.K., Muthukumar, B., Nagarajan, G.: Hybrid intrusion detection system for private cloud: a systematic approach. Procedia Comput. Sci **48**, 325–329 (2015)
6. Sundaram, A.: An introduction to intrusion detection. Crossroads **2**(4), 3–7 (1996)
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. Vieira, K., Schulter, A., Westphall, C., Westphall, C.: Intrusion detection for grid and cloud computing. It Prof. **4**, 38–43 (2009)

9. Tupakula, U., Varadharajan, V., Akku, N.: Intrusion detection techniques for infrastructure as a service cloud. In: 2011 IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing (DASC). IEEE (2011)
10. Kholidy, H.A., Baiardi, F.: CIDS: a framework for intrusion detection in cloud systems. In: 2012 Ninth International Conference on Information Technology: New Generations (ITNG). IEEE (2012)
11. Stolfo, S.J., Salem, M.B., Keromytis, A.D.: Fog computing: Mitigating insider data theft attacks in the cloud. In: 2012 IEEE Symposium on Security and Privacy Workshops (SPW). IEEE (2012)
12. Wang, C., Wang, Q., Ren, K., Lou, W.: Privacy-preserving public auditing for data storage security in cloud computing. In: 2010 Proceedings IEEE INFOCOM, pp. 1–9. IEEE (2010)
13. Diaz-Gomez, P.A., Dean F.H.: Misuse detection-an iterative process vs. a genetic algorithm approach. In: ICEIS, vol. 2 (2007)
14. Govindarajan, M., Chandrasekaran, R.M.: Intrusion detection using neural based hybrid classification methods. Comput. Netw. **55**(8), 1662–1671 (2011)
15. Jeba, L., Rathna, M.: Improving the quality of cloud service by ensemble prediction. Int. J. Appl. Eng. Res. **9**(21), 9323–9326 (2014)
16. Weller-Fahy, D., Borghetti, B.J., Sodemann, A.A.: A survey of distance and similarity measures used within network intrusion anomaly detection. IEEE Commun. Surv. Tutor **17**(1), 70–91 (2014). doi:10.1109/COMST.2014.2336610
17. Bai, Y., Kobayashi, H.: Intrusion detection systems: technology and development. In: 2003 17th International of Advanced Information Networking and Applications AINA 2003, vol. 27(29), pp. 710–715, March 2003
18. Safa, N.S., Ghani, N.A., Ismail, M.A.: An artificial neural network classification approach for improving accuracy of customer identification in e-commerce. Malays. J. Comput. Sci. **27**(3), 171–185 (2014)
19. Rajendran, P.K., Muthukumar, B.: Hybrid intrusion detection algorithm for private cloud. WSEAS Trans. Comput. (2015, Accepted)