Journal of Computer Science 9 (7): 875-882, 2013
ISSN: 1549-3636

doi:10.3844/jcssp.2013.875.882 Published Online 9 (7) 2013 (http://www.thescipub.com/jcs.toc)

The use of Random Forest Classification and K-means Clustering Algorithm for Detecting Time Stamped Signatures in the Active Networks

[1]Kamalanaban Ethala, [1]R. Shesadri and [2]N.G. Renganathan

[1]Department of CSE, Sri Venkateswara University, Tirupathi, India
[1,2]Vel Tech University, Avadi, Chennai

Received 2013-05-24, Revised 2013-06-13; Accepted 2013-07-02

ABSTRACT

In day to day information security infrastructure, intrusion detection is indispensible. Signature based intrusion detection system mechanisms are often available in detecting many types of attacks. But this mechanism alone is not sufficient in many cases. Another intrusion detection method viz K-means is employed for clustering and classifying the unlabelled data. IDS is a special embedded device or relied software package which process of monitoring the events occurring in a computer system or network (WLAN (Wi-Fi, Wimax)) and LAN ((Ethernet, FDDI, ADSL, Token ring) based) and analysing them for sign of possible incident which are violations or forthcoming threats of violations of computer security policies or standard security policies (i.e., DMA acts). We proposed a new methodology for detecting intrusions by means of clustering and classification algorithms. There we used correlation clustering and K-means clustering algorithm for clustering and random forest algorithm for classification. This type of extension establishes a layer which refines the escalated alerts using signature-based correlation. In this study, signature based intrusion detection system with optimised algorithm for better prediction of intrusions has been addressed. Results are presented and discussed.

Keywords: Intrusion Detection System, K-Means, Random Forest, WLAN

1. INTRODUCTION

An Intrusion Detection System (IDS) is an anti-intrusion wall and security layer used to detect and monitor ongoing intrusive activities in Data processing systems and information systems. Traditionally, intrusion detection relies on extensive understanding in knowledge of information and security experts, in particular, on their familiarity with the host or computer system to be protected. To reduce this dependency (Forrester *et al*., 1996), various data-mining methodologies and machine learning techniques, algorithms have been derived, implemented and deployed for intrusion detection. An IDS is a security usually working on a dynamic challenging environment, private zones which powers continuous tuning of the intrusion monitoring logs and intrusion detection model, in order to maintain enough performance and reliability. An intrusion Detection System (IDS) monitors network traffic, suspicious activity and alerts the system or network administrator about the particular event or activities. In some specific cases the IDS may also listen to anomalous or malicious traffic (Forrester *et al*., 1997). Distributed Denial Of Services (DDOS) (Stavrou *et al*., 2005) are by taking action such as blocking or by passing the user temporarily or permanently and source IP address from accessing the network (Lee *et al*., 1997; Lee and Stolfo, 1998; IDSTC, 2013).

There are network based (NIDS) and host based (HIDS) intrusion detection systems. IDS that detect known threats (i.e., looking forward from log table), based on looking for specific behaviour or signatures of

Corresponding Author: Kamalanaban Ethala, Department of CSE, Sri Venkateswara University, Tirupathi, India

much similar to an antivirus software which typically detects ,monitors and protects against malware and viruses. IDS detect based on associating traffic patterns against a fine baseline and looking for anomalies. In this Network Intrusion Detection Systems are placed at an intentional point or points within the network to monitor traffic with inbound and outbound of all devices on the network. Network-based IDS's are mostly passive devices that monitor on-going network activity without adding significant overhead or interfering with network operation. They are easy to secure against attack and may even be undetectable to attackers; they also require little effort to install and use on existing networks. Ideally you would scan all inbound and outbound traffic; however doing so it might create a bottleneck that would impair the overall speed of the network (Warrender *et al*., 1999). On the other hand Host Intrusion Detection Systems will run on individual hosts or devices on the network. HIDS monitors the incoming and outgoing packets from the device only and will alert the user or administrator of suspicious activity is detected (Rawat *et al*., 2005). Yet there are some IDS that simply monitor and alert the administrator when a suspicious activity occurs Similar IDS that perform an arbitrary action or actions in response to a detected threat or intrusion which is known as Signature based IDs which are employed and these are briefly sketched in **Fig. 1**.

1.1. Basic Framework of SIDS

1.1.1. Signature Based Intrusion Detection System

A signature based IDS will monitor packets on the network and compare them against a database of signatures or attributes from known malicious threats. This is similar to the way most antivirus software detects malware **Fig. 2**. The issue is that there will be a lag between a new threat being discovered in the wild and the signature for detecting that threat being applied to your IDS. During that lag time IDS would be unable to detect the new threat (Forrester *et al*., 1997; Wepsi *et al*., 2000; Canvel *et al*., 2003; SSL, 2002a; Perriot and Szor, 2003).

1.2. Anomaly Based Intrusion Detection System

An IDS which is anomaly based will monitor network traffic and compare it against an established baseline. The baseline will identify what is "normal" for that network what sort of bandwidth is generally used, what protocols are used, what ports and devices generally connect to each other-and alert the administrator or user when traffic is detected which is anomalous, or significantly different, than the baseline (Forrester *et al*., 1997; Wepsi *et al*., 2000; Canvel *et al*., 2003; SSL, 2002b; Perriot and Szor, 2003).

With this background the present paper deals with the related works in the Section II, the goals of clustering for intrusion logs in Section III and implementation of the system in Section IV and the algorithm used in Section V and the results and analysis in Section VI. In this the recent review by the present author and from authors may be looked into. In the present communication the study pertaining to algorithms which is more relevant to the present work is provided (KMC, 2013).

The Clustering algorithms may be classified as exclusive clustering, overlapping clustering, hierarchical clustering and probabilistic clustering. In first case the exclusive clustering, data are grouped in a special way with some criterion value and functions, so that one cluster cannot be indexed in another cluster, that is if a certain data belongs to a fixed cluster then it could not be included in another cluster. A simple example of that is shown in the **Fig. 3**, where the split-up of points is achieved by a straight line which passes through two clusters on a bi-dimensional (2D) plane. On the divergent the second type, the overlapping clustering, it use soft computing based fuzzy sets to cluster various data sets, so that each point may belong to two or more clusters with different degrees of membership. In this case, data will be linked to a suitable relationship value (KMC, 2013).

Instead, a hierarchical clustering algorithm is based on the union of two adjacent clusters. The beginning condition is realized by setting every data as a cluster. After a few iterations it reaches the final clusters which are wanted. Finally, the probabilistic clustering used a complete approach which is fully based on probability (KMC, 2013).

In this study we surveyed four of the most used clustering algorithm; they are K-means, Fuzzy C-means, Hierarchical clustering and Mixture of Gaussian. The K-means algorithm is the type of exclusive clustering.

1.3. Space Measure

An important component of a clustering algorithm is the space measure between data points. If the components of the data occurrence vectors are all in the same corporeal units then it is possible that the simple Euclidean distance metric is sufficient to successfully group similar data instances. However, even in this case the Euclidean distance can sometimes be misleading. **Figure 4** illustrates this with an example of the width and height measurements of an object. It also shows, different scaling which can be prime to different clustering's (McClure and Scambray, 2000).

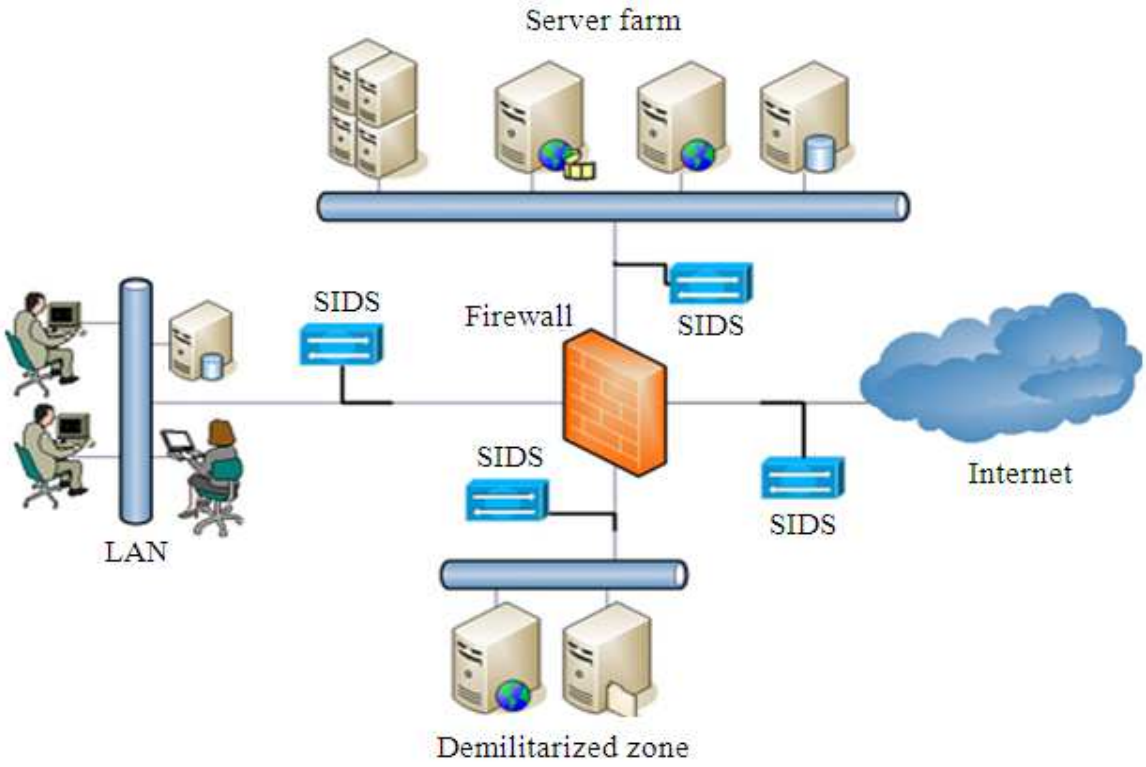

Fig. 1. Basic framework of IDS

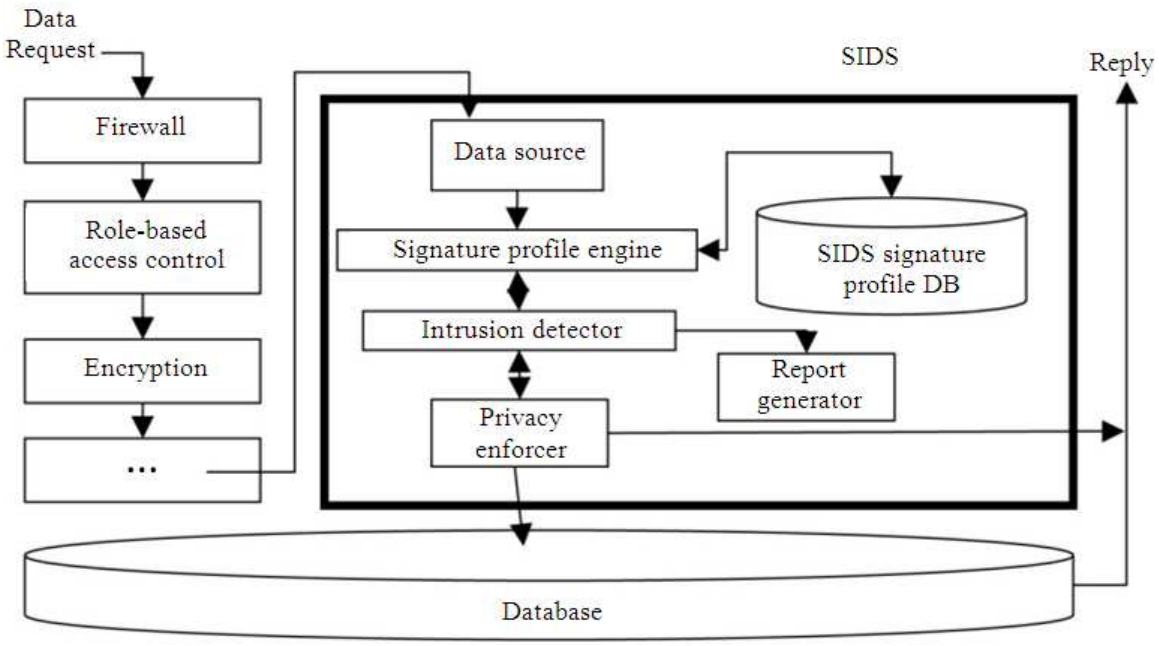

Fig. 2. Basic architecture of SIDS

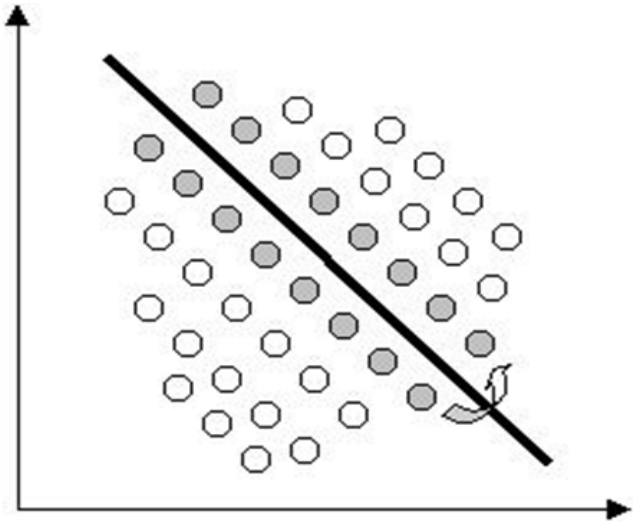

Fig. 3. K-means algorithm

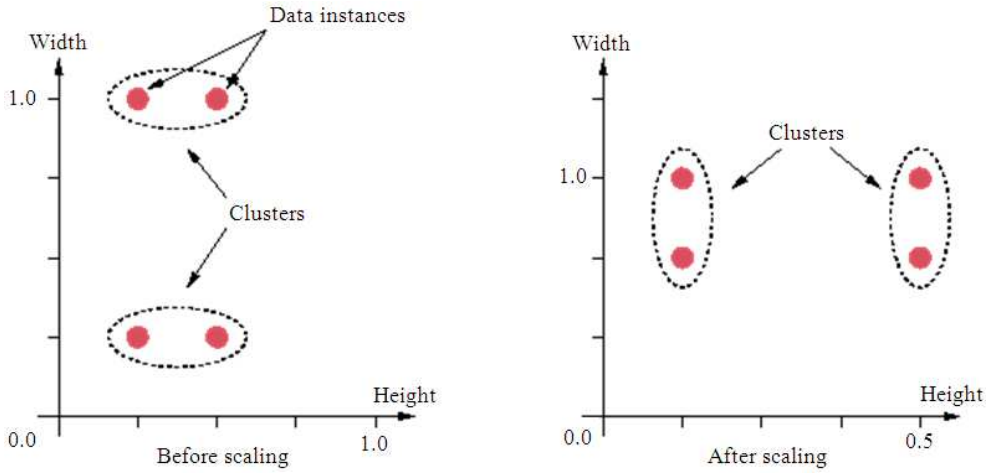

Fig. 4. K-means work model

1.4. The Goals of Clustering for Intrusion Logs

The goal of clustering is to determine the internal grouping and grouping fragmentation in a set of unlabelled data with its inbound and outbound values. Consequently, it is the user which must supply this criterion, in such a way that the result of the clustering criteria will lead to temporal log file generation at run time so that the clustering in intrusion log will suit for our needs. For instance, we could be in finding agents for homogeneous groups (data reduction), in finding usual clusters and describe their unknown properties (usual data types), in finding useful and appropriate grouping (useful data classes) or in finding unusual data objects (outlier detection).

1.5. Classifications

Clustering clearly denotes the unsupervised learning problem. Every other problem deals with finding a structure in a collection of unlabelled data. A loose definition of clustering could be "the process of organizing objects into groups whose members are similar in some way". A cluster is therefore a collection of objects which are "similar" between them and "dissimilar" to the objects belonging to other clusters.

1.6. Random Forest Classification Algorithms

When the training set for the existing tree is drawn by selection with necessary replacement, about 1/3rd of the cases are left out of the trial. This out-of-bab (oob) data is used to get a running unbiased evaluation of the classification error as trees are added to the forest. It is also used to get evaluations of variable importance. After each tree is built, all of the data are run to down the tree and proximities are calculated for each pair of cases. If two cases occupy the same terminal node, their proximity is improved by one. At the end of the run, the proximities are normalizing it by separating the number of trees. Proximities are used in replacing missing data, locating outliers and producing enlightening low-dimensional views of the data.

The main requirements that a k-means clustering algorithm based IDS should satisfy are, scalability, different types of attributes, discovering clusters, arbitrary shape, nominal requirements for domain knowledge to define input parameters, dealing with noise and outliers, insensitivity to order of input sets, high dimensionality, Interpretability and usability.

K-means (KMC, 2013) is the simplest unsupervised learning algorithms that solve the well-known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed a priori. The main idea is to define k centroids, one for each cluster. These centroids should be placed in a sneaky way because of various location causes multiple and different result. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the adjacent centroid. When no point is pending, the first step is completed and an early grouping and group aging is done. At this point we need to re-calculate k new centroids as barycentre of the clusters resulting from the previous step. After we have these k new centroids, a new binding has to be done between the same data set points and the nearest new centroid. A circle has been produced. As a result of this circle we may notice that the k centroids change their location step by step changes are done so that centroids do not move any more. Finally, this algorithm aims at minimizing an objective function, in this case a shaped (Squared) error function (KMC, 2013). The objective function Equation 1:

$$\sum_{j-1}^{k}\sum_{i-1}^{x}\| x(i)^{(j)} - c(j)\|^2 \qquad 1$$

where, $\|x(i)^{(j)}-c(j)\|^2$is a chosen distance measure between a data point x(i)and the cluster centre c(j), is an indicator of the distance of the n data points from their respective cluster centre (KMC, 2013).

1.7. Study of the Deployment

First a temporal file and intrusion log file is created and then the assignment and value are predicted, hence group the dataset values and find the mean value for the group are set and classified the cluster also predicted the error rate or misbehaviour and finally maintained the log table and log file with time stamp.

1.8. Generation of Intrusion Log File

Intrusion log file is generated by means of predicting various datasets from the network by means of some log file, Stats of data, huge data sets. Using the particulars we are indexing it by means of applying K-means algorithm iteratively for whole Data sets. Here pre-processing is done and intrulog file is given as input where the classifier engine verifies for any intrusion or miscellaneous signature and generates report for the sequence of signature within the timestamp.

For this centroid is initially created within the cluster and the value of points are mapped to form the cluster and the k-points into the space which are place where represented by the objects, then predict the nearby centroid and assigned each object to its nearest group, once all objects have been assigned with the values and centroid, are calculated the positions of the K centroids with the values and then when the values have higher centroid and mean correlation based algorithm are used to predict the positions of the k centroids and cluster head and finally repeated steps two and three until the centroids do not have any longer move. This produces a split-up of the objects into groups from which the metric to be minimized can be calculated.

1.9. Algorithm to Draw Random Forest Classification Tree

First Draw Intrulog bootstrap for log files in the main catalog from the original data then second clearly state the bootstrap value with random data set and for each of the bootstrap, grows an unpruned classification or regression tree, with the necessary modification: at each node and then as a third rather than selecting the best split among each and every predictors, randomly sample treelog of the predictors and choose the bestsplit from among those variables and for fourth special case is of random forests obtained when treelog = p, the number of predictors value must be greater in sequence then for

fifth process predict new data by combining the intentions of the Intrulog trees that is majority poll for classification, average for regression and finally an estimate of the error rate can be obtained, based on the training data, as by the following:

```
RandomForestclassification ()
{
    Described Data sets
    mdimension = 4682, nsamplevalue0 = 81,
    nclass = 3, maximumcat = 1,
    testvalue = 0, labelsets = 0, labeltr = 1,

    Set run parameters
    treelog0 = 150, ndsize = 1, jbt = 1000, look1 =
    100, lookclass = 1,
    jclasswt = 0, mdim2nd = 0, mselect = 0, iseed =
    4351,

    Set importance options
    impact = 0, interact = 0, impactn = 0, impactfast
    = 0,

    Set exact proximity computation
    nproximity = 0, nrnn = 5,

    Set options based on exact proximities
    noutlier = 0, nscale = 0, nprot = 0,nintrulog = #

    Replacing missing values
    code = -999, missfill = 0, mfixrepo = 0,

    Graphics
    iviz = 1, isca = 0

    Saving a forest
    Isaverandomf = 0, isavepar = 0, isavefill = 0,
    isaveprox = 0, nitrulog = #, nlog

    Running a saved forest
    Irunrandomf = 0, ireadpar = 0, ireadfill = 0,
    Ireadprox = 0, nintrulog, logfile = #
}
```

1.10. Analysis of Results

Our proposed algorithm results show the reduced false alarm rate with increased performance and analysis of various signatures. Our log file will be updated periodically. **Figure 5** denotes the number of false alarm rate and intrusion (signatures) occurred in a time interval whereas **Fig. 6** clearly denotes the reduced false alarm rate. Here the K means algorithm is revised and iterated in order to find the new signatures. Each Intrulog file is classified using random forest classification and updated. By this any periodical changes and error can be minimized. Error is shown in **Fig. 5** where the optimized algorithm is not effect between Intrulog and logfile. After applying the optimized algorithm where the error is minimized by iteration process to find new signatures and this is done to reduce false alarm rate. This is reflected in the **Fig. 6**, where the difference between intrulog and log file graces the nil error.

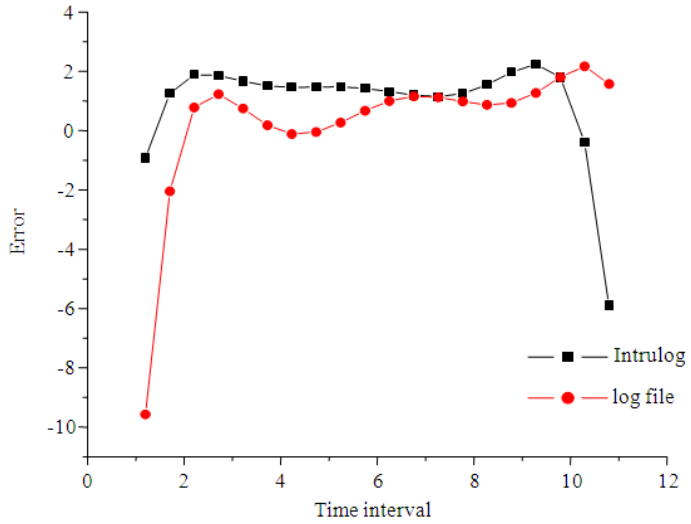

Fig. 5. Signature based intrusion detection system before applying optimised algorithm for prediction of intrusion and miscellaneous behaviour in the network

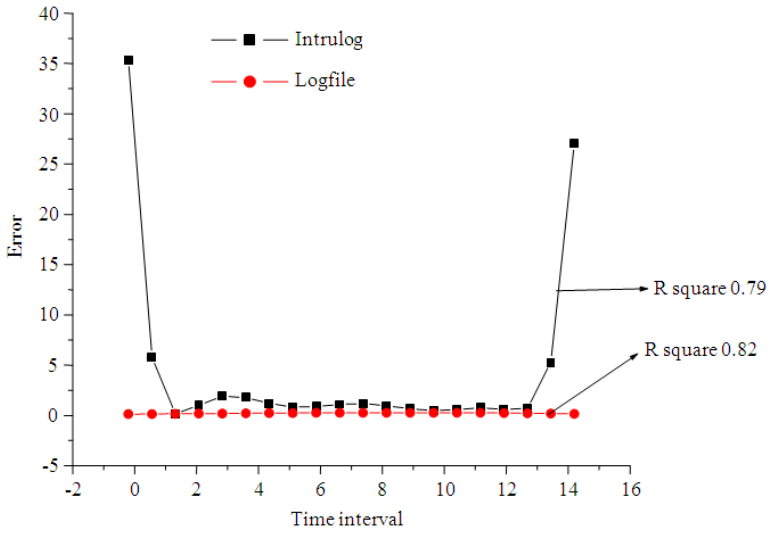

Fig. 6. Signature based intrusion detection system after applying optimised algorithm for prediction of intrusion and miscellaneous behaviour in the network

2. CONCLUSION

This study address about the signature based intrusion detection system with optimised algorithm for better prediction of intrusions and miscellaneous behaviour in the network. Various experiments are carried out with the real time data on network. They have provided evidence that implementation of the proposed algorithm in the monitoring network is realistic. Furthermore, the algorithm builds the database along with intrufile and logfile representing the normal behaviour and miscellaneous behaviour of the profile, which is independent on the traffic load and counter measures. The proposed design is to have optimised the use of K-means clustering algorithm for clustering and random forest algorithm for classification. The scheme manages to avoid false alarms during heavy traffic in networks. This is achieved in the present work and this is indicated as minimum error from the results of employing the optimised algorithm. The graphical representation of simulations exhibits the value of the detection technique. The investigations have considered the intrusion detection delay and the failed session detection error rate indicated by report generator by means of separate log file which acts in independent database. The future enhancement of the present work may also facilitate on identifying and rectifying the problems of cyber terrorism using six degree separation and multi path navigation methodologies.

3. REFERENCES

Canvel, B., A. Hiltgen, S. Vaudenay and M. Vuagnoux, 2003. Password interception in a SSL/TLS channel. Proceedings of the 23rd Annual International Cryptology Conference, Santa Barbara, Aug. 17-21, Springer Berlin Heidelberg, California, USA., pp: 583-599. DOI: 10.1007/978-3-540-45146-4_34

Forrester, S., S.A. Hofmeyr, A. Somayaji and T.A. Longstaff, 1996. A sense of self for UNIX processes. Proceedings of the IEEE Symposium on Research in Security and Privacy, May 6-8, IEEE Xplore Press, Los Alamos, CA., pp: 120-128. DOI: 10.1109/SECPRI.1996.502675

Forrester, S., S.A. Hofmeyr and A. Somayaji, 1997. Computer immunology. Commun. ACM., 40: 88-96. DOI: 10.1145/262793.262811

IDSTC, 2013. Intrusion detection system-types and classification.

KMC, 2013. K-Means Clustering.

Lee, W., S.J. Stolfo and P.K. Chan, 1997. Learning patterns from Unix process execution traces for intrusion detection. Proceedings of the AAA Workshop of AI Methods in Fraud and Risk Management, (AIMFRM' 97), AAAI Press, pp: 50-56.

Lee, W. and S. Stolfo, 1998. Data mining for intrusion detection. Proceedings of the 7th USENIX Association, (USENIXA' 98), pp: 79-94.

McClure, S. and J. Scambray, 2000. Once-promising intrusion detection systems stumble over a myriad of problems. Inform. World, 22: 58-58.

Perriot, F. and P. Szor, 2003. An analysis of the slapper worm exploit. Symantec Security Response, Symantec White Paper.

Rawat, S., V.P. Gulati and A.K. Pujari, 2005. A fast host-based intrusion detection system using rough set theory. Trans. Rough Sets, 3700: 144-161. DOI: 10.1007/11574798_8

SSL, 2002a. OpenSSL servers contain a buffer overflow during the SSL2 handshake process. CERT Vulnerability Note #102795.

SSL, 2002b. OpenSSL servers contain a remotely exploitable buffer overflow vulnerability during the SSL3 handshake process. CERT Vulnerability Note #561275.

Stavrou, A., D.L. Cook, W.G. Morein, A.D. Keromytis and V. Misra *et al*., 2005. WebSOS: An overlay-based system for protecting Web servers from denial of service attacks. Comput. Netw., 48: 781-807. DOI: 10.1016/j.comnet.2005.01.005

Warrender, C., S. Forrest and B. Pearlmutter, 1999. Detecting intrusions using system calls: Alternative data models. Proceedings of the IEEE Symposium on Research in Security and Privacy, May 9-12, IEEE Xplore Press, Oakland, CA., pp: 133-145. DOI: 10.1109/SECPRI.1999.766910

Wepsi, A., M. Dacier and H. Debar, 2000. Intrusion detection using variable-length audit trail patterns. Proceedings of the 3rd International Workshop on the Recent Advances in Intrusion Detection, Oct. 2-4, Springer Berlin Heidelberg, Toulouse, France, pp: 110-129. DOI: 10.1007/3-540-39945-3_8