Adaptive Distributed Intrusion Detection using Hybrid *K*-means SVM Algorithm

Amit Bhardwaj
LMTSOM
Thapar University
Patiala

Parneet Kaur
CSED
Thapar University
Patiala

ABSTRACT
Assuring secure and reliable operation of networks has become a priority research area these days because of ever growing dependency on network technology. Intrusion detection systems (IDS) are used as the last line of defense. Intrusion Detection System identifies patterns of known intrusions (misuse detection) or differentiates anomalous network data from normal data (anomaly detection). In this paper, a novel Intrusion Detection System (IDS) architecture is proposed which includes both anomaly and misuse detection approaches. The hybrid Intrusion Detection System architecture consists of centralized anomaly detection and distributed signature detection modules. Proposed anomaly detection module uses hybrid machine learning algorithm called *k*-means clustering support vector machine (KSVM). This hybrid system couples the benefits of low false-positive rate of signature-based intrusion detection system and anomaly detection system's ability to detect new unknown attacks.

General Terms
Machine Learning, Network Security, Algorithms.

Keywords
Adaptive, Distributed, *k*-means clustering, Intrusion Detection System, Support Vector Machine

1. INTRODUCTION
Files and information stored on systems had to be protected with the introduction of computers. The need for protecting files in computer systems became more evident with the advent of shared systems. Due to recent advances in network technology, computer systems have become even more vulnerable to attacks. Our dependency on network based systems is growing day by day. But protection techniques of such systems have not kept up with the increasing threat. Traditional defense mechanisms such as user authentication, data encryption, avoiding programming loopholes and firewalls are used as the first line of defense against attacks. No combination of technology can protect the system cent percent because systems face novel attacks every other day. So, in this paper we propose Adaptive Distributed Intrusion Detection System that is able to collect data from various hosts to centralized location and identify new attacks as well.

Traditionally, Intrusion detection techniques are categorized as follows: misuse detection and anomaly detection. Misuse detection catches intrusions based on knowledge of known attack patterns, while anomaly detection detects intrusion based on deviation from normal patterns. IDSs based on the misuse detection model generate less false positive alarms and introduce little overhead into the system by detecting only those intrusions which have signatures. Their major drawback, however, is that novel attacks will go undetected until signatures for those intrusions are known to the IDS. IDSs based on anomaly detection model have a better chance of detecting novel intrusions but they are slow due to exhaustive monitoring and use a lot of resources. Also rate of generating false positive alarms is more.

Intrusion Detection Systems can be further categorized as either host based (inspect data from a single host) and network based (examine network traffic from hosts attached to a network). Lastly, IDS is centralized if intrusion data is collected from different hosts or networks and is passed on to a centralized controller component that scrutinizes the information received from each of the monitors [1]. Most of the current IDSs used are distributed ones because Host-based or network-based Intrusion Detection System is almost powerless for complex attacks. The main issue of this kind of system is that it can't identify novel attacks because it is signature based IDS which identifies only well known attack patterns. Data mining methods are used to automate the intrusion detection systems to identify new attacks as well. Most popular way to identify intrusions is by studying the audit data produced by Operating System. Normal system activities are characterized with a profile, which is made by applying mining algorithms to audit data. Abnormal intrusive activities are identified by comparing the current activities with the profile. So in this paper, a feature of adaptation is introduced in it with the help of machine learning algorithm called K means clustering Support Vector Machine. The goal of this paper is to provide a general framework for a hybrid IDS that is both adaptive and distributed.

This work has been divided into three sections. The first section contains machine learning algorithms and the proposed hybrid algorithm. Another section includes proposed framework for IDS using that hybrid algorithm. Finally, the paper is concluded in the last section.

2. PROBLEM DESCRIPTION AND RELATED ISSUES
All Most of the current distributed IDSs are signature based. A major shortcoming of such IDSs is that they can't identify

novel attacks but only well known attack patterns for which signatures are available. To overcome this limitation, IDS is made capable of adapting to the changing attack atmosphere [3]. Data mining methods are used to automate the intrusion detection systems making it anomaly based IDS as well. Short-comings of anomaly based IDS, namely a high false positive rate and the ability to be fooled by a correctly delivered attack are overcome by signature based Distributed architecture. Feature of adaptation is introduced in Distributed IDS with the help of machine learning algorithm. This paper compares two algorithms: SVM and k-means clustering and uses hybrid of the two [4].

2.1 Machine Learning Algorithms

In literature, various anomaly detection systems are developed on the basis of different machine learning techniques. For example, some neural networks, support vector machines, k-means clustering etc are used. In particular, these techniques develop classifiers, which classify the incoming Internet information as normal or intrusion.

2.2 Support Vector Machines

The original SVM algorithm was proposed by Boser, Guyon & Vapnik in 1992. The present standard form (soft margin) was given by Vapnik and Corinna Cortes in 1995 [10].

Support vector machines are supervised learning models that analyze the training data and recognize patterns and produces an inferred function known as classifier (for discrete output) or regression function (for continuous output). The basic SVM studies a set of input data and decides, for each given input, which of two possible classes forms the output. This makes it a non-probabilistic binary linear classifier [12]. The classifier is a function which assigns labels to samples, even those samples which are completely new to the algorithm. Algorithm feeds on previously labeled samples and induces a classifier from them. The key idea in network security is to find useful patterns or features describing user behavior on a system and a set of desired features to construct classifiers. These classifiers are then used to detect anomalies and intrusions from the new coming network traffic [13].

The quality of generalization and ease of training of SVM is way too better than the traditional methods. But the response time of SVM classifiers is still a concern when applied into network intrusion detection. Its limitation is speed and size, both in training and testing [14]. Following are the steps of SVM Algorithm:

- Train SVM on new data set.
  $D=\{(a_i, c_i)| a_i \in R^n, c_i \in \{-1,1\}\}^m_{i=1}$
  where a_i is an n-dimensional real vector and c_i is an indicator of the point a_i belongs to.
- Find the hyperplane separating negative and positive instances of dataset
  $wx-b=0$
  where w is a normal vector to the hyperplane.
- Find shortest distance separating hyperplane to closest positive (negative) data point.
- Find the margin of separating hyperplane $(d+d-)=2/||w||$.
- To get highest confidence classification, maximize the margin. Formulate the linear support vector problem as follows:
  $\text{Max } 1/||w||^2$
  $\text{s.t } c_i(a_i w-b) >= 1 \ \& \ i=[1,m]$
- For separable case when positive and negative data points are linearly separated, they satisfy the following constraints:
  $a_i w-b>=0$, for $c_i=1$,
  $a_i w-b<=0$, for $c_i=-1$
  or they can be combined together into one set:
  $c_i(a_i w-b)-1 >=0$ for all i.
- Solve for w and find the classification

2.3 K-means Clustering Algorithm

The term "*k*-means" was first used by James MacQueen in 1967. The standard algorithm was first introduced by Stuart Lloyd in 1957, though it wasn't published outside Bell labs until 1982 [15].

In data mining, *k*-means clustering is a method of cluster analysis which aims to partition n observations into k clusters in which each observation belongs to the cluster with nearest mean [5]. Simply speaking it is an algorithm to group your objects based on attributes into K groups. This grouping is done by minimizing the sum of squares of distances between data and the corresponding cluster centroid. Aim of K-mean clustering is simply to classify the network data into normal and anomalous.

Following steps shows the demonstration of *k*-means algorithm [5]:

- k initial means are generated within the data domain randomly.
- By associating every observation with the nearest mean k clusters are created.
- The centroid of each cluster becomes the new mean.
- Steps 2 and 3 are repeated until the centroids don't change their position anymore.

This is a very simple and reasonably fast algorithm. It is also efficient in processing large data sets like network traffic. The only difficulty is in comparing the quality of the clusters produced. Another limitation of k-means is that k should be specified in advance. But in Intrusion detection k is set to be two since there are two clusters for normal and anomalous data.

2.4 Comparison of SVM and k-Clustering

SVM is machine learning task of inferring a function from labeled training data. While in *k*-means clustering, machine itself discovers and learn hidden structures present inside unlabeled data [16]. In SVM, predetermined classes are provided. Machine learner's task is to seek patterns and build up mathematical models. In *k*-means clustering, no classification is provided. Machine learner's task is to seek patterns in data and look for likeness among pieces of data so that they can be constituted as a group. No target output labels are present in training and testing datasets of *k*-means clustering in contrast to SVM. The machine simply gets inputs and its job is to learn and differentiate them [11].

3. HYBRID APPROACH: k-SUPPORT VECTOR MEANS

The KSVM algorithm blends the *k*-means clustering technique with SVM and needs another input parameter: the number of clusters. Response time of SVM classifiers can be accelerated by lowering the number of support vectors. *k*-means clustering method is used to gather a data set smaller than the original one to train SVM, which further lowers the number of SVs while maintaining the training accuracy. With

decrease in the number of training examples, computational time of the algorithm falls greatly. There are two approaches for taking advantage of *k*–means clustering algorithm to reduce the number of support vectors used for training the support vector machine. The first approach applies *k*–means clustering to compose a dataset of much smaller size than the actual one. The second approach lowers the number of support vectors by which SVM classifier's decision function is spanned through *k* –means clustering [8].

E[Pr(Error)]<= *E[number of support vectors]/number of training vectors .....(1)*

From inequality (1), it can be deduced that a small number of support vectors will generate a small testing error and also leads to better generalization capability in SVM [9].Successful use of *k*-means requires a cautiously selected distance measure that demonstrates the properties of the clustering task. Designing the distance measure by hand is a tough job. Supervised data is used for training *k*-means even. SVM method that finds a distance measure is used so that *k*-means generates the desired clustering, given the training data sets with desired partitioning [7]. Following are the steps of Adaptive Distributed Intrusion Detection Algorithm using hybrid optimal k-clustering SVM technique:

1. Gather Packets.
2. Apply Misuse Detection Algorithm at nodes.
3. Send remaining packets to centralised anomaly detection agent. Apply k-means clustering Support Vector algorithm.
   - Given a training dataset D containing m data points:
     $D=\{(a_i, c_i)| a_i \in R^n, c_i \in \{-1,1\}\}^m_{i=1}$
     where a_i is an n-dimensional real vector and c_i is an indicator of the class where the point a_i belongs to.
   - Separate the dataset into positive (c=1) and negative (c=2) instances with a hyperplane
     $wx-b=0,$
     where w is normal vector to the hyperplane, x is point of the hyperplane, b is real value, $1/\|w\|$ is perpendicular distance from hyperplane to origin.
   - For separable case when positive and negative data points are linearly separated, they satisfy the following constraints:
     $a_iw-b>=0,$ for $c_i=1,$
     $a_iw-b<=0,$ for $c_i=-1$
     or they can be combined into one set of inequalities:
     $c_i(a_iw-b)-1 >=0$ for all i.
   - Choose w and b to maximize the margin to get highest confidence classification.
     Formulate the linear support vector problem as follows:
     $\text{Max } 1/\|w\|^2$
     $\text{s.t } c_i(a_iw-b) >= 1 \ \& \ i=[1,m]$
   - The resulting two clusters will be assumed as the initial clusters of k clustering algorithm.
   - Set k=2 (for normal and anomalous traffic in training data) initial cluster centres.
   - Assign each packet $x_i \in S$ to the group that has closest centroid
     $s.t \ \| x_i - c_k\| <= \|x_i - c_j\|.$
     Assign x_i to c_k.
   - To get optimum cluster, subset P of the set S should have maximal value of total distance between all instances in the set S.
   - Again calculate the positions of k centroids.
   - Repeat above two steps until centroids no longer move.
4. Nodes sending anomalous packets are informed by centralised node.
5. New anomalous information is updated in Rule Mining Agent which feeds Misuse Detection Agent next time.

4. PROPOSED FRAMEWORK FOR ADAPTIVE DISTRIBUTED INTRUSION DETECTION SYSTEM

The proposed framework is based on the network-based intrusion detection techniques. It is extended from architecture of distributed and adaptive IDS [18, 19]. The architecture (see Figure 1) proposed in this paper is basically composed of five components: Sniffer Agent, Signature based Intrusion Detection Agent, Anomaly based Intrusion Detection Agent, Rule Mining Agent, Signature based Rules Database. Network traffic is captured from different nodes using Sniffer agents such as Wireshark. This information is passed on to Signature based Intrusion Detection Agent which matches the patterns with available rules in Signature based Rules Database. The patterns which correspond to available signatures are declared as intrusion. So, all the known attacks are detected at individual nodes itself [17]. This reduces the burden on centralised node which will now focus on detecting novel attacks. Any suspicious data left is further passed on to centralised node's Anomaly based Intrusion Detection Agent. This agent applies KSVM algorithm which distributes the data into two different clusters: normal and intrusion. The Rule Mining Agent summarizes this information of anomalous data declared by Anomaly Detection Agent and updates Signature based Rules Database with profile of new encountered attack. Misuse Detection Agent is fed from time to time by Signature Database with association rules to update its signatures. If the same attack is faced in future, it will be detected by Signature based Intrusion Detection Agent available at nodes. So, this architecture helps in reducing the burden of resources on one hand while identifying novel attacks on the other [1, 2].

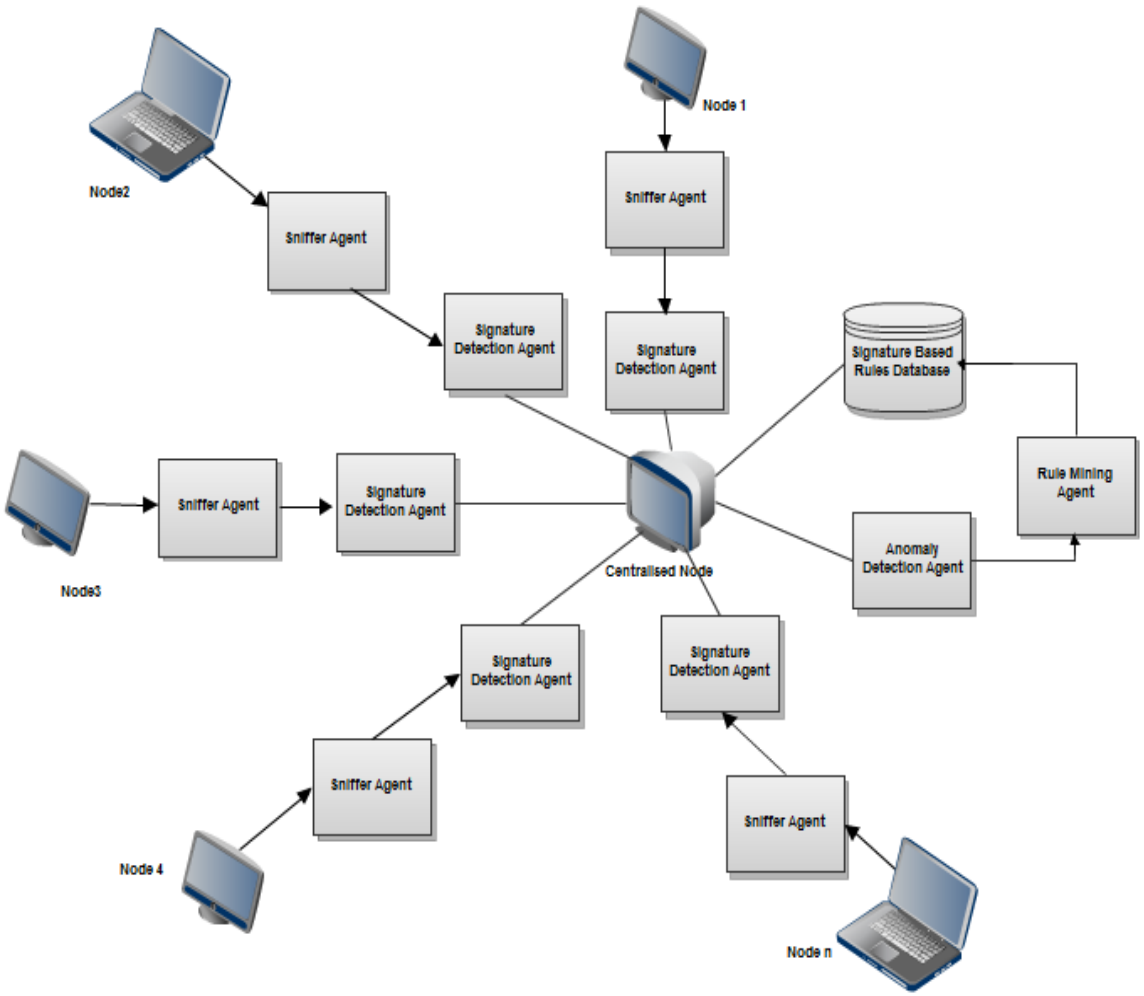

Fig 1: Framework of Adaptive Distributed IDS

5. CONCLUSION

In this paper we proposed a hybrid approach for Intrusion Detection System. We presented adaptive and distributed model for Intrusion Detection System. The method uses the data collected by the sniffer agents of host nodes to detect signature attacks [20]. Novel attacks are detected at the next level by anomaly based centralised node. To make the model adaptive, a hybrid machine learning algorithm called KSVM is used. The algorithm clusters the network traffic into normal and anomalous data [21]. Compared with previous works, our solution has several advantages. First and foremost, our model detects novel attacks. Second, this model significantly reduces the overall load of an IDS system because it distributes the

load on different nodes. Third, *k*-means clustering algorithm reduces the number of Support vectors used which further decreases the computational time [23]. Lastly, high false negative rate of adaptive IDS is taken care of making the some components anomaly based [22].

Future work includes extending the anomaly based component to individual nodes. This will highly increase the overhead and will cause overuse of resources. So in future an approach can be devised which will control the load of resources as well. Also a feature of exchanging suspicious activity among different nodes can be devised so that they communicate directly instead of through centralised node.

6. ACKNOWLEDGMENTS

The authors are highly thankful to Dr Maninder Singh, Thapar University, Patiala without whose guidance and support this work would not have been possible.

7. REFERENCES

[1] Liu, Jianxiao, and Lijuan Li. "A Distributed Intrusion Detection System Based on Agents." Computational Intelligence and Industrial Application, 2008. PACIIA'08. Pacific-Asia Workshop on. Vol. 1. IEEE, 2008.

[2] Huang, Weijian, Yan An, and Wei Du. "A Multi-Agent-based Distributed Intrusion Detection System." Advanced Computer Theory and Engineering (ICACTE), 2010 3rd International Conference on. Vol. 3. IEEE, 2010.

[3] Eskin, Eleazar, Matthew Miller, Zhi-Da Zhong, George Yi, Wei-Ang Lee, and Salvatore Stolfo. "Adaptive model generation for intrusion detection systems." (2000).

[4] Hossain, Mahmood, and Susan M. Bridges. "A framework for an adaptive intrusion detection system

with data mining." 13th Annual Canadian Information Technology Security Symposium. 2001.

[5] Fraley, Chris, and Adrian E. Raftery. "How many clusters? Which clustering method? Answers via model-based cluster analysis." The computer journal 41.8 (1998): 578-588.

[6] (2002) The IEEE website. [Online]. Available: http://www.ieee.org/

[7] Finley, Thomas, and Thorsten Joachims. "Supervised clustering with support vector machines." Proceedings of the 22nd international conference on Machine learning. ACM, 2005.

[8] Jaisankar, N., Swetha Balaji, S. Lalita, and D. Sruthi. "Intrusion Detection System Using K-SVMeans Clustering Algorithm."

[9] Xia, Xiao-Lei, Michael R. Lyu, Tat-Ming Lok, and Guang-Bin Huang. "Methods of decreasing the number of support vectors via K-mean clustering." In Advances in Intelligent Computing, pp. 717-726. Springer Berlin Heidelberg, 2005.

[10] Vishwanathan, S. V. M., and M. Narasimha Murty. "SSVM: a simple SVM algorithm." Neural Networks, 2002. IJCNN'02. Proceedings of the 2002 International Joint Conference on. Vol. 3. IEEE, 2002.

[11] Wang, Jiaqi, Xindong Wu, and Chengqi Zhang. "Support vector machines based on K-means clustering for real-time business intelligence systems." International Journal of Business Intelligence and Data Mining 1, no. 1 (2005): 54-64.

[12] Xie, Lixia, Dan Zhu, and Hongyu Yang. "Research on SVM based network intrusion detection classification." In Fuzzy Systems and Knowledge Discovery, 2009. FSKD'09. Sixth International Conference on, vol. 7, pp. 362-366. IEEE, 2009.

[13] Fang, Xiaozhao, Wei Zhang, Shaohua Teng, and Na Han. "A Research on Intrusion Detection Based on Support Vector Machines." In Communications and Intelligence Information Security (ICCIIS), 2010 International Conference on, pp. 109-112. IEEE, 2010

[14] Shuyue, Wu, Yu Jie, and Fan Xiaoping. "Research on Intrusion Detection Method Based on SVM Co-training." In Intelligent Computation Technology and Automation (ICICTA), 2011 International Conference on, vol. 2, pp. 668-671. IEEE, 2011.

[15] Lakhina, Anukool, Mark Crovella, and Christophe Diot. "Mining anomalies using traffic feature distributions." In ACM SIGCOMM Computer Communication Review, vol. 35, no. 4, pp. 217-228. ACM, 2005.

[16] Ben-Hur, Asa, David Horn, Hava T. Siegelmann, and Vladimir Vapnik. "A support vector clustering method." In Pattern Recognition, 2000. Proceedings. 15th International Conference on, vol. 2, pp. 724-727. IEEE, 2000.

[17] Denning, Dorothy E. "An intrusion-detection model." Software Engineering, IEEE Transactions on 2 (1987): 222-232.

[18] Huang, Ming-Yuh, Robert J. Jasper, and Thomas M. Wicks. "A large scale distributed intrusion detection framework based on attack strategy analysis." Computer Networks 31, no. 23 (1999): 2465-2475

[19] Lee, Wenke, Salvatore J. Stolfo, and Kui W. Mok. "Adaptive intrusion detection: A data mining approach." Artificial Intelligence Review 14, no. 6 (2000): 533-567.

[20] Botía, Juan A., Jorge J. Gómez-Sanz, and Juan Pavón. "Intelligent data analysis for the verification of multi-agent systems interactions." In Intelligent Data Engineering and Automated Learning–IDEAL 2006, pp. 1207-1214. Springer Berlin Heidelberg, 2006.

[21] Arora, A., D. B. Marshall, B. R. Lawn, and M. V. Swain. "Indentation deformation/fracture of normal and anomalous glasses." Journal of Non-Crystalline Solids 31, no. 3 (1979): 415-428.

[22] Axelsson, Stefan. "The base-rate fallacy and the difficulty of intrusion detection." ACM Transactions on Information and System Security (TISSEC) 3, no. 3 (2000): 186-205.

[23] Wen, Yi-Min, and Bao-Liang Lu. "A cascade method for reducing training time and the number of support vectors." In Advances in Neural Networks–ISNN 2004, pp. 480-486. Springer Berlin Heidelberg, 2004.