NIDS For Unsupervised Authentication Records of KDD Dataset in MATLAB

Mis. Bhawana Pillai
M-Tech
LNCT Bhopal
Bhopal, (M.P.) India
bhawanapillai@gmail.com

Mr. Uday Pratap Singh
Asst Prof (CSE) Lnct
Bhopal, (M.P.) India
usinghiitg@gmail.com

Abstract- **Most anomaly based NIDS employ supervised algorithms, whose performances highly depend on attack-free training data. Moreover, with changing network environment or services, patterns of normal traffic will be changed. In this paper, we developed intrusion detection system is to analyses the authentication records and separate UNFEIGNED and fraudulent authentication attempts for each user account in the system. Intrusions are detected by determining outliers related to the built patterns. We present the modification on the outlier detection algorithm. It is important problems to increase the detection rates and reduce false positive rates in Intrusion Detection System. Although preventative techniques such as access control and authentication attempt to prevent intruders, these can fail, and as a second line of defense, intrusion detection has been introduced. Rare events are events that occur very infrequently, detection of rare events is a common problem in many domains. Support Vector Machines (SVM) as a classical pattern recognition tool have been widely used for intrusion detection. However, conventional SVM methods do not concern different characteristics of features in building an intrusion detection system. Also evaluate the performance of K-Means algorithm by the detection rate and the false positive rate. All result evaluate with the new model of KDD dataset. Result generates in ROC Curves and compared both result of K-Means and SVM in Matlab.**

Keywords- ***Anomaly detection; Intrusion Detection; Expectation Maximization;*** *MATLAB;* ***UNSOUND authentication; UNFEIGNED; reduce false.***

I. INTRODUCTION

ECURITY techniques such as authentication and access control have been developed to achieve the objective of computer security namely to prevent unauthorized intruders from accessing and manipulating information. The security administrator is now faced with the problem of selecting suitable IDS for his/her particular computer system. Rapid expansion of computer network throughout the world has made security a crucial issue in a computing environment. Anomalies pattern sometimes exist within tiny or rare classes of similar anomalies. Anomaly-based network intrusion detection is a complex process. The challenge is thus important to identify "rare events" records in data set. As defined in, intrusion detection is "the process of monitoring the events occurring in a computer system or network and analyzing them for signs of intrusions. It is also defined as attempts to compromise the confidentiality, integrity, availability, or to bypass the security mechanisms of a computer or network". Anomaly Intrusion Detection Systems (IDS) aim at distinguishing an abnormal activity from an ordinary one.

Intrusion detection is a critical component of secure information systems. Many approaches have been proposed which include statistical, machine learning, data mining and immunological inspired techniques. Events that may not be actual security violations but those that do not fit in the normal usage profile of a user may be termed as suspicious events. Monitoring suspicious activities may help in finding a possible intrusion.

There are two main intrusion detection systems. *Anomaly intrusion detection system* is based on the profiles of normal behaviors of users or applications and checks whether the system is being used in a different manner.

The second one is called *misuse intrusion detection system* which collects attack signatures, compares a behavior with these attack signatures, and signals intrusion when there is a match. It is often impossible to analyze the vast amount of whole data, but one has to focus the analysis on an important portion of the data such as using some criteria, only the classes of interest can be selected for analysis or processing while the rest is rejected. This paper suggests the use rough set as a dimensionality reduction technique to avoid this information loss.

The theory of rough sets has been specially designed to handle data imperfections same as in fuzzy logic. Rough sets remove superfluous information by examining attribute dependencies. It deals with inconsistencies, uncertainty and incompleteness by imposing an upper and a lower approximation to set membership. Rough sets estimates the relevance of an attribute by using attribute dependencies regarding a given decision class. It achieves attribute set covering by imposing a discernibility relation With the tremendous growth of network-based services and sensitive information on networks, the number and the severity of network-based computer attacks have significantly increased. Although a wide range of security technologies such as information encryption, access control, and intrusion prevention can protect network-based systems, there are still many undetected intrusions. Thus, Intrusion Detection Systems (IDS) play a vital role in network security. Network Intrusion Detection Systems (NIDS) detect attacks by observing various network activities, while Host-based Intrusion Detection Systems (HIDS) detect intrusions in an individual host.

To overcome the limitations of supervised anomaly based systems, a number of IDS employ unsupervised approaches. Unsupervised anomaly detection does not need attack-free training data. It detects attacks by determining unusual activities from data under two assumptions: The majority of activities are normal. Attacks statistically deviate from normal activities. The unusual activities are outliers that are inconsistent with the remainder of data set. Thus, outlier detection techniques can be applied in unsupervised anomaly detection. Actually, outlier detection has been used in a number of practical applications such as credit card fraud detection, voting irregularity analysis, and severe weather prediction.

II. Proposed Technique

Data presented to algorithm is generated by picking one of two Gaussians at random and then sampling from the selected distribution. If each Gaussian describes on of two users – UNFEIGNED and fraudulent, trying to authenticate, knowing from which Gaussian each sample of our data originated would completely solve our ID problem. Gaussian type distributions are assumed here for both UNFEIGNED and fraudulent user, So what are the hidden variables in this problem? Well, if we knew which sample in our set is generated by which distribution we could easily solve the problem. It would then be easy to calculate sample mean and variance for each distribution. All that would be left in this situation would be to somehow classify the new samples (i.e. new authentication attempts) as members of one or the other Gaussian.

The *k*-Means clustering is a classical clustering algorithm. After an initial random assignment of example to *k* clusters, the centers of clusters are computed and the examples are assigned to the clusters with the closest centers. The process is repeated until the cluster centers do not significantly change. Once the cluster assignment is fixed, the mean distance of an example to cluster centers is used as the score. Using the *k*means clustering algorithm, different clusters were specified and generated for each output class. There are two problems that are inherent to *k*-Means clustering algorithms. The first is determining the initial partition and the second is determining the optimal number of clusters.

Algorithm 1. k-means

Step 1: Choose *k* cluster centers to coincide with *k* randomly-chosen patterns or *k* randomly defined points inside the hyper volumn containing the pattern set.

Step 2: Assign each pattern to the closest cluster center.

Step 3: Recomputed the cluster centers using the current luster memberships.

Step 4: If a convergence criterion is not met, go to step 2. Typical convergence criteria are: no (or minimal) reassignment of patterns to new luster centers, or minimal decrease in squared error

In this experiment, we use a standard dataset the raw data used by the KDD Cup 1999 intrusion detection contest. This database includes a wide variety of intrusions simulated in a military network environment that is a common benchmark for evaluation of intrusion detection techniques. Test data use filename “corrected.gz” contains a total of 38 training attack types. It consists of approximately 300,000 data instances, each of which is a vector of extracted feature values from a connection record obtained from the raw network data gathered during the simulated intrusion and is labeled normal or a certain attack type. The 41 features can be divided into three groups; the first group is the basic feature of individual TCP connections, the second group is the content feature within a connection suggested by domain knowledge, and the third group is the traffic feature computed using a two-second time window. The distribution of attacks in the KDD Cup dataset is extremely unbalanced. Some attacks are represented with only a few examples, e.g. the phf and ftp_write attacks, whereas the smurf and neptune attacks cover millions of records. In general, the distribution of attacks is dominated by probes and denial-of-service attacks; the most interesting and dangerous attacks, such as compromises, are grossly under represented.

The data set has 41 attributes for each connection record plus one class label. There are 24 attack types, but we treat all of them as an attack group. A data set of size N is processed. The nominal attributes are converted into linear discrete values (integers). After eliminating labels, the data set is described as a matrix X, which has N rows and m=14 columns (attributes).

III. Test Results and Analysis

In This thesis we are take 1000 sample data and two algorithms K-Means, and SVM and there result are given below

First we taken SVM algorithm and sample data 1000 so we get result in following:-

a) False Positive rates. b) True Positive Rates

Receiver operating characteristic curve

We summarize our experimental results to detect intrusions using the unsupervised outlier detection technique over the KDD’99 dataset. We first describe the datasets used in the experiments. Then we evaluate our approach and discuss the results.

Under the sponsorship of Defense Advanced Research Projects Agency (DARPA) and Air Force Research Laboratory (AFRL), MIT Lincoln Laboratory has collected and distributed the datasets for the evaluation of computer network intrusion detection systems [20, 21]. The DARPA dataset is the most popular dataset used to test and evaluate a large number of IDSs. The KDD’99 dataset is a subset of the DARPA dataset prepared by Sal Stofo and Wenke Lee [25]. The data was preprocessed by extracting 41 features (e.g., protocol type, service, and flag) from the tcpdump data in the 1998 DARPA dataset. The KDD’99 dataset can be used without further time-consuming preprocessing and different IDSs can compare with each other by working on the same dataset. Therefore, we carry out our experiments on the

KDD'99 dataset. The full training set, one of the KDD'99 datasets, has 4,898,431 connections, which contains attacks. The attacks in the dataset fall into four categories [26]: DoS (Denial of Service), R2L (unauthorized access from a remote machine), U2R (unauthorized access to root privileges), and probing. The dataset is labeled by type of attacks. Since our approach is unsupervised, the dataset does not satisfy the needs of our experiments. We must remove the labels that indicate types of attacks from the dataset.

To generate new datasets for our experiments, we first separate the dataset into two pools according to the labels. One includes normal connections. Another includes attacks. Then, we remove all the labels from the pools. However, we need the data labeled by service to build patterns of services,

So we use service feature in the dataset as label. As a result, all the data contains 40 features and is labeled by service. For our experiments, we choose five most popular network services: ftp, http, pop, smtp, and telnet.

Five different types of data were chosen with 40 attributes each [27]. The data contain 24 attack types which are classified into four categories. They are Denial of Service (DOS), unauthorized access from a Remote Machine (URM), unauthorized access to Local Super user (ULS) and Probing and Surveillance (PAS). Denial of service (DOS) is a class of attack where an attacker makes a resource too busy to handle authorized request and in turn deny access to the authorized users. URM is a class of attack where an attacker exploits the vulnerability of the machine by sending packets to the machine, to gain illegal access as a user. In the case of ULS an attacker starts with gaining access to the account of a normal user and then exploits the systems vulnerability. PAS is a class of attack where an attacker scans a network to know the vulnerabilities and exploits them. The 40 variables are given in Table 5.1 the variables from 24 to 40 are modeled using normal distribution. The variables 8 and 9 are modeled using they are numerically viable. All the data are normalized between 0 and 1. A clustering algorithm is used for classifying them into five classes namely, NORMAL, PAS, DOS, URM and ULS. The true positive rates and false positive rates for are obtained using the formula

a) True positive rate = (positives correctly classified)/ (total positives)
b) False positive rate = (total negatives – negatives incorrectly classified)/ (Total negatives).

TABLE1. VARIABLE

Variable Name	Variable Name
Duration	Is-guest _login
Protocol Type	Count
Service	Srv_count
Flag	Serror_rate
Src_bytes	Srv_serror_rate
Dst_bytes	Rerror_rate
Wrong fragment	Srvr_rerror_rate
Urgent	Same_srv_rate
Hot	Diff_srv_rate
Num_failed _logins	Srv_diff_host_rate
Logged_ in	Dst_host_count
Num_compromized	Dst_host_srv_count
Root_shell	Dst_host_same_srv_rate
Su_attempted	Dst_host_diff_srv_rate
Num_root	Dst_host_same_src_port_rate
Num_file_creations	Dst_host_srv_diff_host_Rate
Num_shells	Dst_host_serror_rate
Num_access_files	Dst_host_srv_serror_rate
Num_outbound_cmds	Dst_host_rerror_rate
Is_host_login	Dst_host_srv_rerror_rate

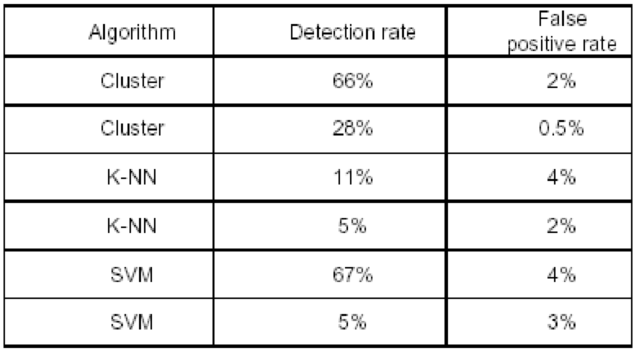
A. Evaluation and discussion from K-Means

We carry out the first experiment over the attack dataset. We first optimize the parameters of K-Means algorithm by feeding the dataset into the NIDS. The NIDS builds patterns of the network services with different values of the parameters.

With the optimized parameters, we build the patterns of the network services. Over the built patterns, the NIDS calculates the Iteration of each connection. Since the attacks are injected at the beginning of the dataset, the figure shows the Iteration of the attacks is much higher than most of normal activities. Some normal activities also have high Iteration. That leads to false positives. The NIDS will raise an alert if an Iteration of a connection exceeds a specified threshold.

We evaluate the performance of K-Means algorithm by the detection rate and the false positive rate. The detection rate is the number of attacks detected by the system divided by the number of attacks in the dataset. The false positive rate is the number of normal connections that are misclassified as attacks divided by the number of normal connections in the dataset. We can evaluate the performance by varying the threshold of outlier-ness.

TABLE 2 THE PERFORMANCE OF EACH ALGORITHM OVER THE KDD'99 DATASET [1]



Algorithm	Detection rate	False positive rate
Cluster	66%	2%
Cluster	28%	0.5%
K-NN	11%	4%
K-NN	5%	2%
SVM	67%	4%
SVM	5%	3%

In intrusion detection, ROC (Receiver Operating Characteristic) curve is often used to measure performance of IDSs. The ROC curve is a plot of the detection rate against the false positive rate. Fig. 1 plots ROC curve to show the relationship between the detection rates and the false positive rates over the dataset. The result indicates that K-Means algorithm can achieve a high detection rate with a low false positive rate. Compared to other unsupervised anomaly based systems [2, 10], our system provides better performance over the KDD'99 dataset while the false positive rate is low.

A. *Evaluation and discussion from K-Means Vs SVM*

We carry out the first experiment over the attack dataset. We first optimize the parameters of K-Means and SVM algorithm by feeding the dataset into the NIDS. The NIDS builds patterns of the network services with different values of the parameters.

In intrusion detection, ROC (Receiver Operating Characteristic) curve is often used to measure performance of IDSs. The ROC curve is a plot of the detection rate against the false positive rate. Fig. 1 plots ROC curve to show the relationship between the detection rates and the false positive rates over the dataset.

The result indicates that K-Means algorithm can achieve a high detection rate with a low false positive rate. Compared to other unsupervised anomaly based systems [2, 10], our system provides better performance over the KDD'99 dataset while the false positive rate is low.

TABLE 3 COMPARISON OF ROC CURVES SVM ~ K-MEAN

	AUC	SE [a]	95% CI [b]
Test_Data_In_SVM	0.718	0.0739	0.689 to 0.746
Test_Data_K_mean	0.766	0.0512	0.738 to 0.791

Show that the detection rate is reduced significantly when the false positive rate is low. Although our experiments are carried out under different conditions, Fig. 1 shows that our K-Means algorithm still provides relatively higher detection rates when the false positive rates are low. For example, the detection rate is 97.9%

TABLE 4 PAIR WISE COMPARISON OF ROC CURVES

Test_Data_In_SVM ~ Test_Data_K_mean	
Difference between areas	0.0470
Standard Error [c]	0.0631
95% Confidence Interval	-0.0766 to 0.171
z statistic	0.745
Significance level	P = 0.4560

To evaluate our system under different number of attacks, we carry out the experiments over attack dataset. Fig. 1 plots the ROCs for each dataset using comparison of ROC curves. The result shows that the performance tends to be reduced while increasing number of attacks.

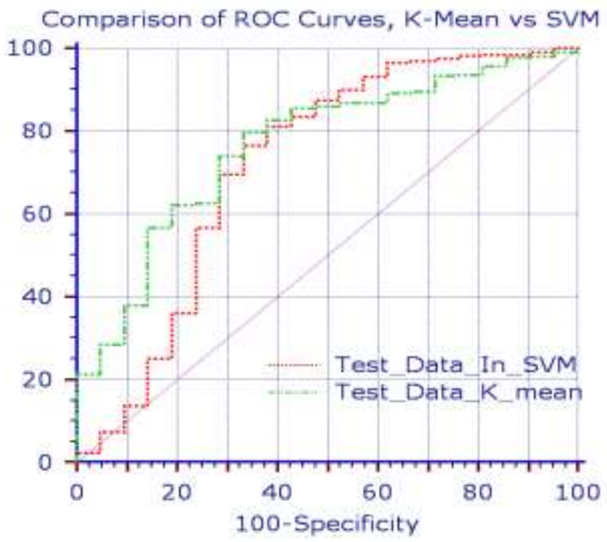


Figure 1. The comparison of ROC curves for the datasets

IV. CONCLUSION

In this Paper' goal was to provide the scientific evidence that one-class SVMs and K-Means algorithm can be regarded as suitable method for detecting intrusions in flow-based network data The performance of K-Means algorithm is comparable to that of other reported unsupervised anomaly detection approaches. Especially, our approach achieve higher detection rate when the false positive rate is low. It is more significant for NIDSs, since high false positive rate will make NIDSs useless. Due to high complexity of the unsupervised anomaly detection algorithm, low detection speed performance of the approach makes real time detection impossible. However, the approach can detect novel intrusions without attack-free training data. The detected novel intrusions can be used to train real time supervised misuse detection systems. Therefore, the trained misuse detection systems can detect the novel intrusions in real time.

The results also show that the performance tends to be reduced with increasing number of attack connections. That is a problem of unsupervised systems. Some attacks (e.g., DoS) produce a large number of connections, which may undermine an unsupervised anomaly detection system. To overcome the problem, we will incorporate both anomaly based and misuse based approaches into the NIDS in the future. Misuse approach can detect known attacks. By removing known attacks, the number of attacks can be reduced significantly in datasets for unsupervised anomaly detection. Misuse detection has high detection rate with low false positive rate. Anomaly detection can detect novel attacks to increase the detection rate. Therefore, combining misuse and anomaly detection can improve the overall performance of the NIDS.

REFERENCES

[1] Bhawana Pillai, Mr. Vineet Rechhariya Network Intrusion Detection For Unsupervised Authentication RecordsIn Matlab icices-2011

[2] E. Eskin, A. Arnold, M. Prerau, L. Portnoy, and S. Stolfo, "A Geometric Framework for Unsupervised Anomaly Detection: Detecting Intrusions in Unlabeled Data", Applications of Data Mining in Computer Security, Kluwer, 2002.

[3] Rasheda Smith, Alan Bivens, Mark Embrechts, Chandrika Palagiri, and Boleslaw Szymanski, “Clustering Approaches for Anomaly Based Intrusion Detection”, Walter Lincoln Hawkins Graduate Research Conference 2002 Proceedings, New York, USA, October 2002.

[4] Susan M.Bridges, and Rayford B. Vaughn, “Fuzzy Data Mining and Genetic Algorithms Applied to Intrusion Detection”, Proceedings of the National Information Systems Security Conference (NISSC), Baltimore, MD, October, 2000.

[5] Alan Bivens, Mark Embrechts, Chandrika Palagiri, Rasheda Smith, and Boleslaw Szymanski, “Network-Based Intrusion Detection Using Neural Networks”, Artificial Neural Networks In Engineering, St. Louis, Missouri, November 2002.

[6] Q.A. Tran, H. Duan, and X. Li, “One-class Support Vector Machine for Anomaly Network Traffic Detection”, The 2nd Network Research Workshop of the 18th APAN, Cairns, Australia, 2004.

[7] A. Lazarevic, L. Ertoz, A. Ozgur, J. Srivastava & V. Kumar, “A Comparative Study of Anomaly Detection Schemes in Network Intrusion Detection”, Proceedings of Third SIAM Conference on Data Mining, San Francisco, May 2003.

[8] J. Zhang and M. Zulkernine, “Network Intrusion Detection Using Random Forests”, Proc. of the Third Annual Conference on Privacy, Security and Trust, St. Andrews, New Brunswick, Canada, October 2005.

[9] Kingsly Leung and Christopher Leckie, “Unsupervised Anomaly Detection in Network Intrusion Detection Using Clusters”, Australasian Computer Science Conference, Newcastle, NSW, Australia, 2005.

[10] M. Ramadas, S. Ostermann and B. Tjaden, “Detecting Anomalous Network Traffic with Self-Organizing Maps”, RAID, 2003.

[11] Ian H. Witten, and Eibe Frank, Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations, Morgan Kaufmann publishers, October 1999.

[12] V. Barnett and T. Lewis, Outliers in Statistical Data, John Wiley, 1994.

[13] C. T. Lu, D. Chen, and Y. Kou, “Algorithms for Spatial Outlier Detection”, Proceedings of 3rd IEEE International Conference on Data Mining, Melbourne, Florida, November 2003.

[14] K. Tan, K. Killourhy, and R. Maxion, “Undermining an anomaly based intrusion detection system using common exploits”, RAID, Zurich, Switzerland, Oct. 2002.

[15] Snort, Network Intrusion Detection System, http://www.snort.org.

[16] L. Breiman, “Random Forests”, Machine Learning 45(1):5–32, 2001.

[17] Lan Guo, Yan Ma, Bojan Cukic, and Harshinder Singh, “Robust Prediction of Fault-Proneness by Random Forests”, Proceedings of the 15th International Symposium on Software Reliability Engineering (ISSRE'04), pp. 417-428, Brittany, France, November 2004.

[18] Bogdan E. Popescu, and Jerome H. Friedman, Ensemble Learning for Prediction, Doctoral Thesis, Stanford University, January 2004.

[19] J. D. Cannady. An adaptive neural network approach to intrusion detection and response. PhD thesis, Nova Southeastern University, 2000.

[20] Yimin Wu, High-dimensional Pattern Analysis in Multimedia Information Retrieval and Bioinformatics, Doctoral Thesis, State University of New York, January 2004.

[21] M. Mahoney and P. Chan, “An Analysis of the 1999 DARPA/Lincoln Laboratory Evaluation Data for Network Anomaly Detection”, Proceeding of Recent Advances in Intrusion Detection (RAID), Pittsburgh, USA, September 2003.

[22] J. D. Cannady. Next generation intrusion detection: Autonomous reinforcement learning of network attacks. In NISSC ’00: Proc. 23rd National Information Systems Security Conference, 2000.

[23] MIT Lincoln Laboratory, DARPA Intrusion Detection Evaluation, http://www.ll.mit.edu/IST/ideval/, MA, USA.

[24] Daniel Barbarra, Julia Couto, Sushil Jajodia, Leonard Popyack, and Ningning Wu, “ADAM: Detecting Intrusions by Data Mining”, Proceedings of the 2001 IEEE, Workshop on Information Assurance and Security, T1A3 1100 United States Military Academy, West Point, NY, June 2001.

[25] WEKA software, Machine Learning, http://www.cs.waikato.ac.nz/ml/weka/, The University of Waikato, Hamilton, New Zealand.

[26] Charles Elkan, “Results of the KDD'99 Classifier Learning”, SIGKDD Explorations 1(2): 63-64, 2000

[27] KDD’99 datasets, The UCI KDD Archive, http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, Irvine, CA, USA, 1999

[28] Ting-Fan Wu, Chih-Jen Lin, and Ruby C. Weng, “Probability Estimates for Multi-class Classification by Pairwise Coupling”, The Journal of Machine Learning Research, Volume 5, December 2004.

[29] D. Sarjon and Mohd Noor Md Sap, “Association Rules using Rough Set and Association Rule Methods”, Proceedings of 7th Pacific Rim International Conference on Artificial Intelligence (PRICAI-02), Tokyo, Japan, August 18-22, 2002, pp. 238-243.