ISSN 2250-3153

Enhancing the features of Intrusion Detection System by using machine learning approaches

Swati Jaiswal, Neeraj Gupta, Hina Shrivastava

Abstract- The IDS always analyze network traffic to detect and analyze the attacks. The attack detection methods used by these systems are of two types: anomaly detection and misuse detection methods. Intrusion detection (ID) is a type of security management system for computers and networks. An ID system gathers and analyzes information from various areas within a computer or a network to identify possible security breaches, which include both intrusions and misuse. An Intrusion detection system is designed to classify the system activities into normal and abnormal. ID systems are being developed in response to the increasing number of attacks on major sites and networks. Intrusion detection is the act of detecting unwanted traffic on a network or a device. Several types of IDS technologies exist due to the variance of network configurations. In this paper, we provide you information about the methods that uses a combination of different machine learning approaches to detect a system attacks.

Index Terms- machine learning, IDS, neural network.

I. INTRODUCTION

Intrusion detection is the act of detecting unwanted traffic on a network or a device. An IDS can be a piece of installed software or a physical appliance that monitors network traffic in order to detect unwanted activity and events such as illegal and malicious traffic, traffic that violates security policy, and traffic that violates acceptable use policies. Several types of IDS technologies exist due to the variance of network configurations. Each type has advantages and disadvantage in detection, configuration, and cost. The IDSs are very useful for detecting, identifying and pursuing internet intruders.

The IDSs always analyze network traffic to detect and analyze the attacks. The attack detection methods used by these systems are of two types: anomaly detection and misuse detection methods [2]. Most misuse detection methods are based on attack signature detection which are determined by network and security experts. After the signatures are determined, they are compared with the network input traffic in order to detect and recognize new attacks. Misuse detection methods have shown to be very efficient [3]. On the contrary, anomaly detection methods always compare suspicious and normal traffics. In fact, they are to prevent unexpected attacks. In order for anomaly detection to take place, they learn normal and abnormal traffic features and then they detect new attacks based on the degree of deviation from normal traffic [4]. Different methods are used for analyzing and learning the traffic.

A Network Intrusion Detection System (NIDS) is one common type of IDS that analyzes network traffic at all layers of the Open Systems Interconnection (OSI) model and makes decisions about the purpose of the traffic, analyzing for suspicious activity. Most NIDSs are easy to deploy on a network and can often view traffic from many systems at once. Network behavior anomaly detection (NBAD) views traffic on network segments to determine if anomalies exist in the amount or type of traffic. Segments that usually see very little traffic or segments that see only a particular type of traffic may transform the amount or type of traffic if an unwanted event occurs. Host-based intrusion detection systems (HIDS) analyze network traffic and system-specific settings such as software calls, local security policy, local log audits, and more. In the following, a combinatory method based on decision trees, KNN algorithms and neural networks is introduced to take advantage of each of these methods in detecting both anomaly and misuse attacks.

Decision Trees are excellent tools for helping you to choose between several courses of action. They provide a highly effective structure within which you can lay out options and investigate the possible outcomes of choosing those options. Decision tree learning, used in statistics, data mining and machine learning, uses a decision tree as a predictive model which maps observations about an item to conclusions about the item's target value. More descriptive names for such tree models are classification trees or regression trees. Different data mining algorithms are also good solutions for attack detections [5] which is already discussed by Quinlin, J. R. Decision trees one of the most powerful and effective ways of detecting attacks especially in anomaly detections [6, 7,8].

There are many ways by which attack signatures can be found such as decision tree, different types of neural networks and SVM [10]. In the following, a combinatory method based on decision trees, KNN algorithms and neural networks is introduced to take advantage of each of these methods in detecting both anomaly and misuse attacks.

This survey paper contains III section, in which section II deals with the review of related works. Section III focuses on the methods proposed by Hadi Sarvari, Mohammad Mehdi Keikha [1] in IDSCML2. Section IV elaborates the overall conclusion of the paper.

II. REVIEW OF RELATED WORK

The IDSs 1 are one of the important security parts connected to internet networks since there are many ways to violate the security of networks. The IDSs are very useful for detecting, identifying and pursuing internet intruders. The IDSs always analyze network traffic to detect and analyze the attacks. The IDSs 1 are one of the important security parts connected to internet networks since there are many ways to violate the security of networks. The IDSs are very useful for detecting, identifying and pursuing internet intruders. The IDSs always

analyze network traffic to detect and analyze the attacks. As defined in Sun, Merrill, and Peterson (2001), top-down learning goes from explicit to implicit knowledge. They also had better memory recall performance. Typing performance after the top-down learning process was faster than the initial performance of the control group. Three top down learning methods - alphabetical grouping, word based-grouping, and word generation – were designed to provide users with constructive visual search strategies and improve their memory of the keyboard layout during learning sessions.

The second type of techniques is machine learning techniques. These techniques are used when there is no primary knowledge about the pattern of the data. That is why they are sometimes called bottom-up methods. Since our method is based on machine learning methods, we are going to have a brief look at some IDSs which were done by using machine learning methods.Machine learning, a branch of artificial intelligence, is a scientific discipline concerned with the design and development of algorithms that allow computers to evolve behaviors based on empirical data, such as from sensor data or databases. A major focus of machine learning research is to automatically learn to recognize complex patterns and make intelligent decisions based on data; the difficulty lies in the fact that the set of all possible behaviors given all possible inputs is too large to be covered by the set of observed examples (training data).

Nadjaran and Kahani did attack detection by using fuzzy neural networks and fuzzy deduction [12]. First, they use a number of fuzzy neural networks for primary classification and then by applying fuzzy deduction to primary Classifications output, they see whether the system activity flow is normal or it is an attack. If it is an attack, they specify its type.

Considering the two types of attack features as continuous and discrete, Weiming recognizes discrete features with the weak classification of the discrete features and continuous features with the weak classification of the continuous features [13]. After the two steps, he combines the results of continuous and discrete classification so as to use the advantages of continuous and discrete features for attack detections at the same time. The article [14] classifies the input data by using different types of tree classifications and then selects the best classification of the given sample by using Ant Colony. The SSGBML system works in two phases to detect the sample group. Primarily it uses Steady State GA to detect and classify the rules of each class and after analyzing their fitness, it increases the accuracy of this classification by using a Zeroth classifier and getting feedback from the setting. It also tries to improve the classification quality through the mechanism of learning from the environment [15].

The LAMSTAR IDS has increased the ability to learn a variety of attacks and has decreased the learning time of the neural networks by using the sample selector and classifier algorithms. This system selects samples with more information to increase the system accuracy and deletes the irrelevant features of the attack detection and classification in the learning time so that it can decrease the learning time of SOM neural networks used to detect attacks [16]. Supervised machine learning is the search for algorithms that reason from externally supplied instance to produce general hypotheses, which then make predictions about future instances. In other words, the goal of supervised learning is to build a concise model of the distribution of class labels in terms of predictor features. The resulting classifier is then used to assign class labels to the testing instances where the values of the predictor features are known, but the value of the class label is unknown.

III. USE OF KDD99 CUP & IDSCML SYSTEM

In earlier papers decision tree were used to find out the attacks. Decision trees one of the most powerful and effective ways of detecting attacks especially in anomaly detections [6, 7, 8].

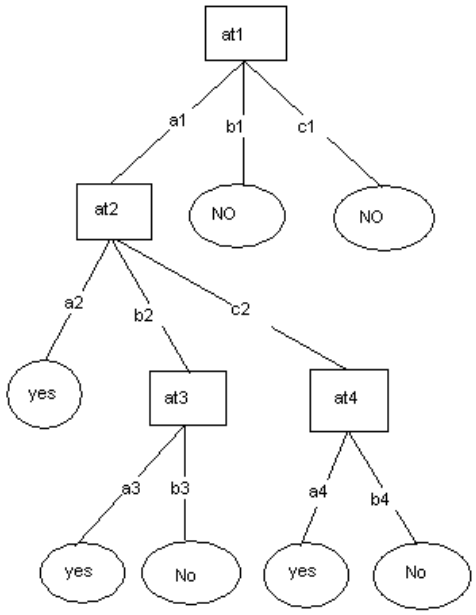


Fig 1: Decision Tree

Table 2: Training Set

at1	at2	at3	at4	Class
a1	a2	a3	a4	yes
a1	a2	a3	b4	yes
a1	b2	a3	a4	yes
a1	b2	b3	b4	no
a1	c2	a3	a4	yes
a1	a2	a3	b4	no
a1	a2	a3	b4	yes

But in some cases decision tree is alone not capable of finding all types of attacks. Data mining (which is the analysis step of Knowledge Discovery in Databases) focuses on the discovery of *unknown* properties on the data.

KDD 99 is a standard source of data for evaluating IDSs which appeared in 1990. The data includes the US air force local network together with a variety of simulated attacks. In this collection, every sample is equal to a connection. A connection is a sequence of TCP packages which starts and ends at a specific time with the flow of data from the source IP address to the destination IP. TCP is a transport layer protocol used by applications that require guaranteed delivery. TCP establishes a full duplex virtual connection between two endpoints. Each endpoint is defined by an IP address and a TCP port number. 41 features were defined for each connection in this collection, which are divided into four major categories namely: primary

features of TCP protocol, content features, time-based and host-based traffic features. Every connection has a label which determines whether it is normal or it is one of the defined attacks. This collection contains 24 different known attacks and 14 unknown ones which have been included in experimental data. TCP provides a communication service at an intermediate level between an application program and the Internet Protocol (IP). That is, when an application program desires to send a large chunk of data across the Internet using IP, instead of breaking the data into IP-sized pieces and issuing a series of IP requests, the software can issue a single request to TCP and let TCP handle the IP details. Due to network congestion, traffic load balancing, or other unpredictable network behavior, IP packets can be lost, duplicated, or delivered out of order. TCP detects these problems, requests retransmission of lost data, rearranges out-of-order data, and even helps minimize network congestion to reduce the occurrence of the other problems. Once the TCP receiver has reassembled the sequence of octets originally transmitted, it passes them to the application program. Thus, TCP abstracts the application's communication from the underlying networking details.

The present attacks are divided into four categories as follows: U2R, R2L, DOS and PROBE. Figure3 illustrates the data distribution for each attack out of one million new KDD 99 packs of data. As it can be seen in Figure3, the major problem with this data set is lack of balance between the numbers of samples in each class.

Table 3: Data Distribution in Training Data set

TYPE	Number Of Training Samples	Percentage Of Samples in Training Set
Normal	812814	75.61
DOS	247287	23.0
Probe	13860	1.29
R2L	979	0.09
U2R	52	0.00

In KDD99cup the problem of balance and unbalanced occur. So in paper [1] IDSCML SYSTEM is used. In this the combination of decision tree and KNN is used. These two methods are separately applied to the data set. 1NN and DT were successful in some detection cases on their own right. By taking advantage of both 1NN and DT, the outputs of 1NN and DT were combined by using a BP neural network in figure 1, which improved the results to some extent. The outcome of this combination turned out to be much better than that of one or both 1NN and DT.

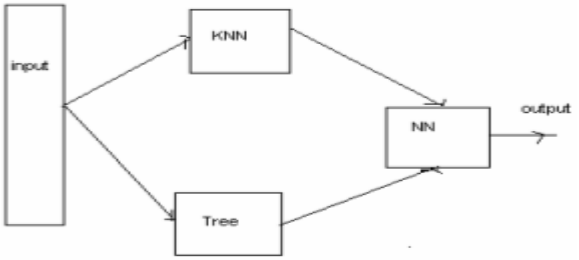
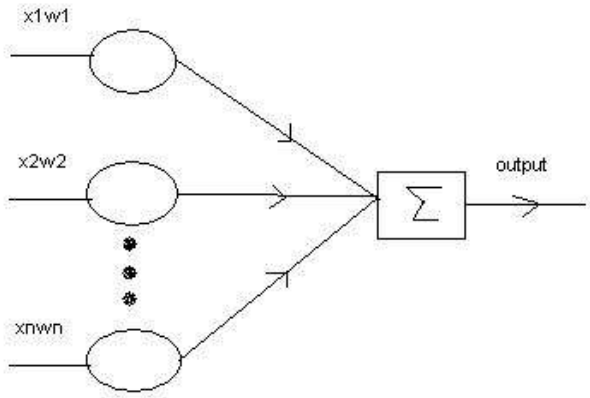


Fig 4: Combining Tree with KNN

Kohonen neural networks are used in data mining process and for knowledge discovery in databases. As all neural networks it has to be trained using training data. The Kohonen neural network library is a set of classes and functions to design, train and calculates results from Kohonen neural network known as self organizing map. The Kohonen neural network differs both in how it is trained and how it recalls a pattern. The Kohohen neural network does not use any sort of activation function. Further, the Kohonen neural network does not use any sort of a bias weight. Output from the Kohonen neural network does not consist of the output of several neurons.When a pattern is presented to a Kohonen network one of the output neurons is selected as a "winner". This "winning" neuron is the output from the Kohonen network. Often these "winning" neurons represent groups in the data that is presented to the Kohonen network.



To continue with, 2NN and 3NN were separately applied to the data and it was revealed that they do better than 1NN and DT. 2NN was added to the combination in figure 1 which again resulted in a better attack detection than the 1NN and DT combination.

Based on these results, it could be predicted that the more the number of classifiers increases, the more the accuracy of the system would be. This motivated us to use other classifiers too. So SVM was added to the system and we gained better results as it has been shown in the seventh row of table 3. Later, LVQ and BP neural networks were added but surprisingly, the results got worse or remained constant. So we revised our prediction and came to the conclusion that the accuracy of the system increases by adding the number of classifiers but this improvement stops at one point and remains constant from that point on.

According to table 1, the U2R and R2L class samples are much fewer than other classes and therefore their detection percentage is less than that of the others, too. This is called the unbalanced data problem with input data for different classes. In order for this problem to be eliminated in the combinatory model, we tried to generate data with features similar to the features of these two classes so that the system could recognize the members of these two classes by observing more samples of their members. In fact, by this way we increased the accuracy of the system in testing step. An entrance was created for a neural network by adding first level classifiers namely, DT, SVM, 1NN, 2NN and 3NN and the results of every classification were saved. Now these results were used as the neural network entrance for the final recognition of samples.

Since the class 4 (R2L) and the class 3 (U2R) samples of data were few and the neural network does not get enough training for the features of these two classes, we have to generate some data for these classes so that it can recognize their attack with more accuracy. Vectors like (*,*,*,*, 3), (*,*,*, 3,*), (*,*, 3,*,*), (*, 3,*,*,*), (3,*,*,*,*) were used to generate data in the third class. Note that *= {1 or 2 or 3 or 4 or 5} , meaning that * can be any class and the favorable result for these data is the third class. It also indicates that only if one classifier recognizes the third class, we indirectly tell the neural network that the proper class for the data is the third class. In this way, we can boost the coefficients of the third class samples. The purpose of adding these data is to make the third class be the final result whenever a classifier recognizes the third class. This addition of the data may generate data which is not in the input data. In other words, we have added an unknown behavior to the system data to make the third class be the final result whenever a classifier recognizes the third class. Class 4 (R2L) also has little data, so we do the same thing for it. But in this case *= {1 or 2 or 4 or 5}.

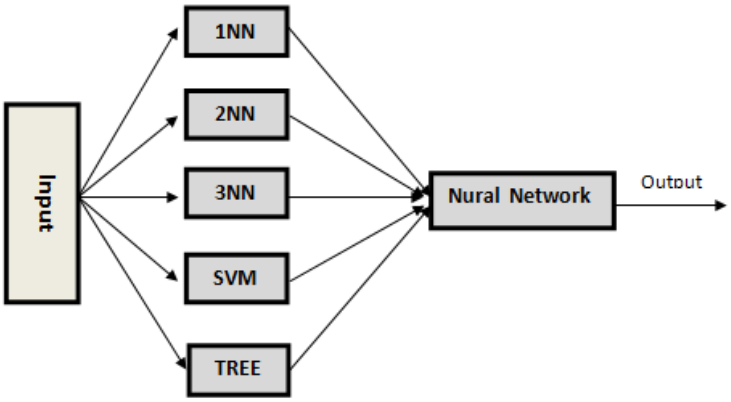


Fig 5: Several Combinatory methods

Here a table is given which indicates the performance of different methods used to find out the attacks.

	Decision Trees	Neural Networks	Naïve Bayes	KNN	SVM	Rule learners
Accuracy in general	**	***	*	**	****	**
Speed of learning with respect to number of attributes and the number of instances	***	*	****	****	*	**
Speed of classification	****	****	****	*	****	****
Tolerance to missing values	***	*	****	*	**	**
Tolerance to irrelevant attributes	***	*	**	**	****	**
Tolerance to redundant attributes	**	**	*	**	***	**
Tolerance to highly interdependent attributes (e.g. parity problems	**	***	*	*	***	**
Dealing with discrete/binary/continuous attributes	****	***	***	***	**	***
Tolerance to noise	**	**	***	*	**	*
Dealing with danger of overfitting	**	*	***	***	**	**
Attempts for incremental learning	**	***	****	****	**	*
Explanation ability/transparency of knowledge/classifications	****	*	****	**	*	****

The data was given to the system several times in order to have better neural network training. Another advantage of this system is that it works well with both continuous and discrete features. It changes the continuous features to their proper and corresponding discrete features and then they are grouped into different classifiers.

IV. CONCLUSION

We compared the results of IDSCML with other systems in table V. KDD 99 contains 24 different known attacks in the training data and 14 unknown attacks in the testing data. This article proposes a combinatory system. Primarily, it was concluded that the system accuracy increases by increasing the

number of classes. Then it was revealed that this increase continues to a point at which it stops and remains constant. One of the features of KDD 99 is that their samples are much fewer than those of other classes and since machine learning-based systems do not learn the features of these two classes, their detection accuracy is also much less than other classes. In fact, unknown features have been introduced to the system. The table is given below that indicates the comparison results of different methods with IDSCML.

Table 5: Comparison result of Some Systems with Idscml

Method & Systems	Normal	DOS	Probe	U2R	R2L
MSSGBML [14]	96.32	97.6	37.72	28.85	83.93
ESCIDS[11]	98.2	99.5	84.1	14.1	31.5
Multi classifier [15]	96.34	97.3	88.7	29.8	9.6
PNrule [16]	99.5	96.9	73.2	6.6	10.7
IDSCML (Proposed method)	99.92	98.2	93.22	44.44	93.21

REFERENCES

[1] Hadi Sarvari, Mohammad Mehdi Keikha, Improving the Accuracy of Intrusion Detection Systems by Using the Combination of Machine Learning Approaches, University of Isfahan, 978-1-4244-7896-5/10/$26.00_c 2010 IEEE.

[2] O. Depren, M. Topallar, E. Anarim, and M. KemalCiliz, An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks. Expert Systems with Applications, Volume 29, Issue 4, pp. 713-722, November 2005.

[3] J. H. Lee, S. G.Sohn, J. H. Ryu, and T. M. Chung Effective Value of Decision Tree with KDD99 Intrusion Detection Data sets for Intrusion Detection System. In International Conference on Advanced Communication Technology, 17-20, pp.1170-1175, Feb 2008.

[4] R. A. Kemmerer, and G. Vigna, Intrusion detection: A brief history and overview. IEEE Security and Privacy Magazine (supplement toComputer), 35(4), vol. 35 no. 4, pp. 27-30, April 2002

[5] Quinlin, J. R. Decision Trees and Decision Making. In IEEE Trans. on System (Man and Cybernetic), 20(2), 1990, 339-346.April 1990.

[6] Abbes, T., Bouhoula, A., and Rusinowitch, M. Protocol Analysis in Intrusion Detection Using Decision Tree. In Proceeding of the International Conference on Information Technology: Coding and Computing (ITCC'04) 1, pp .404-408, Vol 1, Las Vegas, April 2004.

[7] C. Kruegel, and T.Toth, Using Decision Trees to Improve Signature Based Intrusion Detection. In Proceeding of the 5th Recent Advances in Intrusion Detection 2003 (RAID2003), LNCS2820,173- 191.Volume 2, pp.404, April 2004 .

[8] V. H. Garcia, R.Monroy, and M. Quintana, Web Attack Detection Using ID3. IFIP International Federation for Information Processing,. Springer Santiago, Chile, Vol. 218, pp 323-332, 2006.

[9] L. Kuang, and M. Zulkernine, an Anomaly Intrusion Detection Method Using the CSI-KNN Algorithm. Proceeding of the 2008 ACM symposium on applied computing. Brazil, pp. 921-926, 2008.

[10] Y. Li, B. Fang, L. Guo, and Y. Chen, Network Anomaly Detection Based on TCM-KNN Algorithm. Proceeding of the 2nd ACM Symposium on Information, computer and communications security,pp. 13-19, 2007.

[11] Z. S. Pan, S. C. Chen, G. B. Hu, and D. Q. Zhang, Hybrid Neural Network and C4.5 for Misuse Detection. Proceeding of the Second International Conference on Machine Learning and Cybernetics (Xi'an, 2003). 20463-20467.IEEE in china, pp.2-5 November 2003.

[12] A. Nadjaran, and M. Kahani, A new approach to intrusion detection based on an evolutionary soft computing model using neuro-fuzzyClassifiers, Computer Communications 30(2007), pp. 2201-2212.July 2007.

[13] W. Hu, and S. Maybank, Ada Boost-Based Algorithm for Network Intrusion Detection. In IEEE TRANS. ON SYSTEMS MAN ANDCYBERNETICS PARTB: CYBERNETICS, VOL. 38, NO. 2, APRIL 2008.

[14] L. P.Rajeswari, A. Kannan, and R. Baskaran, An Escalated Approach to Ant Colony Clustering Algorithm for Intrusion Detection System. In International Conference on Distributed Computing and Networking. 393-400.Springer, Kolkata, India, 5-8,.Lecture Notes in Computer Science 4904, January 2008

[15] W. S. Sharafat,, and R. Naoum, Development of Genetic-based Machine Learning for Network Intrusion Detection (GBML-NID).World Academy of Science, Engineering and Technology, pp. 20-24, 2009.

[16] M. R. Sabhnani, , and G. Serpen, Application of machine learning algorithms to KDD intrusion detection data set within misuse detection context. In Proceeding of International Conference on Machine Learning: Models, Technologies and Applications (Las Vegas, Nevada, USA, 2003). Pp.209-215., CSREA, June 2003, InPress.

[17] S. B. Kotsiantis, Supervised Machine Learning: A Review of Classification Techniques, Department of Computer Science and Technology University of Peloponnese, Greece End of Karaiskaki, 22100, Tripolis GR.

AUTHORS

First Author – Swati Jaiswal, AP, SAMCET, swatijaiswal26@gmail.com

Second Author – Neeraj Gupta, AP, SAMCET, gupta_neeraj3108@yahoo.co.in

Third Author – Hina Shrivastava, SIRT, BHOPAL, India hina.shrivastava1509@gmail.com