Adaptive Network Intrusion Detection Learning: Attribute Selection and Classification

Dewan Md Farid, Jérôme Darmont, Nouria Harbi, Huu Hoa Nguyen, Mohammad Zahidur Rahman

Adaptive Network Intrusion Detection Learning: Attribute Selection and Classification

Dewan Md. Farid[1], Jerome Darmont[1], Nouria Harbi[1], Nguyen Huu Hoa[1], and Mohammad Zahidur Rahman[2]

[1]Labratoire ERIC, Universite Lumiere Lyon 2 – 5 av. Pierre Mendes, France – 69676 BRON Cedex
cedmfarid@hotmail.com, jerome.darmont@univ-lyon2.fr, nouria.harbi@univ-lyon2.fr, nhhoa@eric.univ-lyon2.fr
[2]Dept. Of CSE, Jahangirnagar University, Savar, Dhaka-1342, Bangladesh
rmzahid@juniv.edu

***Abstract*—**In this paper, a new learning approach for network intrusion detection using naïve Bayesian classifier and ID3 algorithm is presented, which identifies effective attributes from the training dataset, calculates the conditional probabilities for the best attribute values, and then correctly classifies all the examples of training and testing dataset. Most of the current intrusion detection datasets are dynamic, complex and contain large number of attributes. Some of the attributes may be redundant or contribute little for detection making. It has been successfully tested that significant attribute selection is important to design a real world intrusion detection systems (IDS). The purpose of this study is to identify effective attributes from the training dataset to build a classifier for network intrusion detection using data mining algorithms. The experimental results on KDD99 benchmark intrusion detection dataset demonstrate that this new approach achieves high classification rates and reduce false positives using limited computational resources.

***Keywords*—**Attributes selection, Conditional probabilities, information gain, network intrusion detection.

I. Introduction

INTRUSION detection systems (IDS) have become an integral part of today's information security infrastructures. IDS aims at detecting unauthorized activities that attempt to compromise the confidentiality, integrity, and availability of computer systems or resources [1]. The concept of IDS was first introduced by James P. Anderson in 1980 [2] and later formalized by Dr. Dorothy Denning in 1986 [3]. The explosive increase in the number of computer networks and the use of Internet in every organization has led to an increase of attacks, not only from external intruders but also from internal intruders. A network IDS observes alerts, but most of the alerts are false positives, which are not related to the security incidences. One of the most important problems for intrusion detection is effective attributes selection from training dataset, because the volume of dataset that an IDS needs to examine is very large even for a small network and contains large number of attributes. The complex relationships exist among the attributes of datasets, which are difficult to analysis and harder to detect suspicious behavior patterns [4], [5].

Computer security deals with the protection of data and the computing resources and is commonly associated with confidentiality, integrity, and availability [6]. Intrusion is a violation of information security policy. Modern computer systems apply identification, authentication, and authorization for managing information security systems. Currently two types of intrusion detection models are using by IDS for detecting intrusions: anomaly-based detection model and misuse-based detection model. The anomaly-based intrusion detection model identifies new attacks by analyzing the strange behaviors from the normal behaviors in computer systems. On the other side, the misuse-based intrusion detection model performs simple pattern matching to match a pattern corresponding to a known attack type in the database of computer systems. In complex classification domains of intrusion detection, attributes of dataset may contain false correlations, which hamper the process of detecting intrusions. Some attributes of dataset may be redundant, because the information they add is contained in other attributes. Some extra attributes can increase computation time, and can have an impact on the detection accuracy. In IDS effective attributes selection improves the classification accuracy by searching for the subset of attributes [7]. Also some data in the dataset may not be useful for intrusion detection and thus can be eliminated before learning. In this paper, we proposed a new learning approach using naïve Bayesian classifier and ID3 algorithm to address the problem of effective attributes selection and classification for network intrusion detection.

Recently, data mining algorithms are using to build IDS that classifies network connections for detecting intrusions [8]. Lee developed a data mining framework for constructing attributes using domain-specific knowledge to built IDS [9]. Fan built IDS with a data mining technique that is a comprehensive study of cost-sensitive learning using classifier ensembles [10]. Maloof and Michalski investigate incremental learning algorithms and applied to intrusion detection [11]. They underline the significance of symbolic representation language and human understandability of background knowledge and criticized a neural network approach. Another example of the application of symbolic learning to intrusion detection using

user signatures is presented [12].

The remainder of this paper is organized as follows. Section II provides a review of intrusion detection learning. Section III describes our proposed approach for network intrusion detection learning. Section IV provides the experimental analysis based on KDD99 benchmark intrusion detection dataset [13]. Finally, Section V presents our conclusions.

II. Intrusion Detection Learning

A. Intrusion Detection Systems

Originally intrusion detection system (IDS) was implemented for host-based IDS that located in servers to examine the internal interfaces [19], but with the evolution of computer networks the focus gradually shifted towards network-based IDS. Context Sensitive String Evaluation (CSSE) is one of the Host-based IDS (HIDS) for defending attacks in applications with extremely low false-positives [14]. CSSE uses an instrumented execution environment (such as PHP or Java Virtual Machine) and therefore has access to all necessary contexts required to detect and more importantly prevent attacks. The context is provided by the metadata, which describes the fragments of the output expression that requires checking and examining the intercepted call to the API function. CSSE uses contextual information to check the unsafe fragments for syntactic content. Depending on the mode of CSSE it can raise an alert and prevent the execution of the dangerous content (both intrusion detection and prevention). Currently CSSE is available as research-prototype IDS for the PHP platform [15], [16]. Snort is an open source network intrusion detection and prevention system (NIDPS) capable of performing packet logging and real-time traffic analysis of IP networks. Snort was written by Martin Roesch and is now developed by Sourcefire. Snort performs protocol analysis, content searching/matching, and is commonly used to actively block a variety of attacks [17]. Most of the current attacks happen at higher layers: transport (TCP/UDP) or application (HTTP, RPC) layers and Snort uses so-called preprocessors which perform stream reassembly and normalization of higher-level protocols. To detect an attack targeting a web server the preprocessors normalize the IP-level traffic, TCP state machine emulation and stream reassembly, HTTP-level normalization, defragmentation, and Unicode decoding.

B. Attributes Selection from Dataset

Effective input attributes selection from intrusion detection datasets is one of the important research challenges for constructing high performance IDS. Irrelevant and redundant attributes of intrusion detection dataset may lead to complex intrusion detection model as well as reduce detection accuracy. This problem has been studied during the early work of W.K. Lee [5], research on KDD99 benchmark intrusion detection dataset, where 41 attributes were constructed for each network connection. The attribute selection methods of data mining algorithms identify some of the important attributes for detecting anomalous network connections. Attributes selection in intrusion detection using data mining algorithms involves the selection of a subset of attributes d from a total of D original attributes of dataset, based on a given optimization principle. Finding a useful attribute subset is a form of search. Ideally, attribute selection methods search through the subsets of attributes, and try to find the best one among the completing 2^N candidate subsets according to some evaluation function. Therefore, building IDS based on all attributes is infeasible, and attributes selection becomes very important for IDS.

In KDD99 intrusion detection dataset, there are total 494021 examples in the 10% training dataset. The KDD99 dataset contains 22 different attack types that could be classified into four main categories namely Denial of Service (DoS), Remote to User (R2L), User to Root (U2R) and Probing. There are 41 attributes for each network connection that have either discrete values or continuous values. The attributes in KDD99 dataset can be divided into three groups. The first group of attributes is the basic features of network connection, which include the duration, prototype, service, number of bytes from source IP addresses or from destination IP addresses, and some flags in TCP connections. The second group of attributes in KDD99 is composed of the content features of network connections and the third group is composed of the statistical features that are computed either by a time window or a window of certain kind of connections. The attributes selection in KDD99 dataset has been widely used as a standard method for network-based intrusion detection learning, and it was found that all 41 attributes of KDD99 dataset are not the best ones for intrusion detection learning. Therefore the performance of IDS may be further improved by studying new attribute selection methods [18].

C. Classifier Construction

Classifier construction is another important research challenge to build efficient IDS. Nowadays, many data mining algorithms have become very popular for classifying intrusion detection datasets such as decision tree, naïve Bayesian classifier, neural network, genetic algorithm, and support vector machine etc. However, the classification accuracy of most existing data mining algorithms needs to be improved, because it is very difficult to detect several new attacks, as the attackers are continuously changing their attack patterns. Anomaly network intrusion detection models are now using to detect new attacks but the false positives are usually very high. The performance of an intrusion detection model depends on its detection rates (DR) and false positives (FP). DR is defined as the number of intrusion instances detected by the system divided by the total number of the intrusion instances present in the dataset. FP is an alarm, which rises for something that is not really an attack. It is preferable for an intrusion detection model to maximize the DR and minimize the FP. For DR, we can modify the objective function to 1-DR. Therefore classifier construction for IDS is another technical challenge in the field of data mining.

III. Proposed Learning Algorithm

Given a training data $D = \{t_1,...,t_n\}$ where $t_i = \{t_{i1},...,t_{ih}\}$ and the training data D contains the following attributes $\{A_1, A_2,...,A_n\}$ and each attribute A_i contains the following attribute values $\{A_{i1}, A_{i2},...,A_{ih}\}$. The attribute values can be discrete or continuous. Also the training data D contains a set of classes $C = \{C_1, C_2,...,C_m\}$. Each example in the training data D has a particular class C_j. The algorithm calculates the information gain for each attributes $\{A_1, A_2,...,A_n\}$ from the training data D.

$$Info(D) = -\sum_{j=1}^{m} \frac{freq(C_j, D)}{|D|} \log_2 \left(\frac{freq(C_j, D)}{|D|} \right) \quad (1)$$

$$Info(T) = \sum_{i=1}^{n} \frac{|T_i|}{|T|} \inf o(T_i) \quad (2)$$

$$Information\ Gain\ (A_i) = Info(D) - Info(T) \quad (3)$$

Then the algorithm chooses one of the best attributes A_i among the attributes $\{A_1, A_2,...,A_n\}$ from the training data D with highest information gain value, and split the training data D into sub-datasets $\{D_1, D_2,...,D_n\}$ depending on the chosen attribute values of A_i. The algorithm then estimates the prior and conditional probabilities for each sub-dataset $D_i = \{D_1, D_2,...,D_n\}$ and classifies the examples of sub-dataset D_i using their respective probabilities. The prior probability $P(C_j)$ for each class is estimated by counting how often each class occurs in the dataset. For each attribute A_i the number of occurrences of each attribute value A_{ij} can be counted to determine $P(A_i)$. Similarly, the conditional probability $P(A_{ij}|C_j)$ for each attribute values A_{ij} can be estimated by counting how often each value occurs in the class in the dataset. For classifying an example in the dataset, the prior and conditional probabilities generated from the dataset are used to make the prediction. This is done by combining the effects of the different attribute values from the example. Suppose the example e_i has independent attribute values $\{A_{i1}, A_{i2},...,A_{ip}\}$, we know $P(A_{ik} | C_j)$, for each class C_j and attribute A_{ik}. We then estimate $P(e_i | C_j)$ by

$$P(e_i | C_j) = P(C_j) \prod_{k=1 \to p} P(A_{ij} | C_j) \quad (4)$$

To classify an example in the dataset, the algorithm estimates the likelihood that e_i is in each class. The probability that e_i is in a class is the product of the conditional probabilities for each attribute value with prior probability for that class. The posterior probability $P(C_j | e_i)$ is then found for each class and the example classifies with the highest posterior probability for that example. The algorithm will continue this process until all the examples of sub-datasets or sub-sub-datasets are correctly classified. When the algorithm correctly classifies all the examples of all sub/sub-sub-datasets, then the algorithm terminates and the prior and conditional probabilities for each sub/sub-sub-datasets are preserved for future classification of unseen examples. The main procedure of proposed algorithm is described as follows.

Algorithm

Input: Training Dataset D

Output: Intrusion Detection Model

Procedure:

1. Calculate the information gain for each attributes $A_i = \{A_1, A_2,...,A_n\}$ from the training data D using equation (3).
2. Choose an attribute A_i from the training data D with the maximum information gain value.
3. Split the training data D into sub-datasets $\{D_1, D_2,...,D_n\}$ depending on the attribute values of A_i.
4. Calculate the prior $P(C_j)$ and conditional probabilities $P(A_{ij}|C_j)$ of each sub-dataset D_i.
5. Classify the examples of each sub-dataset D_i with their respective prior and conditional probabilities.
6. If any example of sub-dataset D_i is misclassified then again calculate the information gain of attributes of sub-dataset D_i, choose the best attribute A_i with maximum information gain value from sub-dataset D_i, split the sub-dataset D_i into sub-sub-datasets D_{ij} and again calculate the prior and conditional probabilities for each sub-sub-dataset D_{ij}. Finally, classify the examples of sub-sub-datasets using their respective prior and conditional probabilities.
7. Continue this process until all the examples of sub/sub-sub-datasets are correctly classified.
8. Preserved all the prior and conditional probabilities for each sub-dataset D_i or sub-sub-dataset D_{ij} for future classification of unseen examples.

IV. Experimental Results

A. Intrusion Detection Data Stream

The experiment was carried out on a real data stream called "intrusion detection dataset", which has been used in the Knowledge Discovery and Data Mining (KDD) 1999 Cup competition [13]. In KDD99 dataset the input data flow contains the details of the network connections, such as protocol type, connection duration, login type etc. Each data sample in KDD99 dataset represents attribute value of a class in the network data flow, and each class is labeled either as normal or as an attack with exactly one specific attack type. In total, 41 features have been used in KDD99 dataset and each connection can be categorized into five main classes (one normal class and four main intrusion classes: probe, DOS, U2R, R2L). There are 22 different types of attacks that are grouped into the four main types of attacks (probe, DOS, U2R, R2L) tabulated in Table 1.

TABLE I.
DIFFERENT TYPES OF ATTACKS IN KDD99 DATASET

4 Main Attack Classes	22 Attacks Classes
Probing	ipsweep, nmap, portsweep, satan
Denial of Service (DOS)	back, land, eptune, pod, smurt, teardrop
User to Root (U2R)	buffer_overflow, perl, loadmodule, rootkit
Remote to User (R2L)	ftp_write, guess_passwd, imap, multihop, phf, spy, warezclient, warezmaster

The experimental setting is the same as the one used in the KDD99 Cup [13], taking 10% of the whole real raw data stream (494021 data samples) for training and 311029 data samples for testing. Table 2 shows the number of training and testing examples for each class in KDD99 dataset.

TABLE II.
NUMBER OF EXAMPLES IN KDD99 DATASET

Attack Types	Training Examples	Testing Examples
Normal	97277	60592
Probing	4107	4166
Denial of Service	391458	237594
User to Root	52	70
Remote to User	1126	8606
Total Examples	494020	311028

B. Experimental Analysis

Our experiments have two phases namely learning and classifying training data and then classifying the testing data. In the first phase, important attributes from training data of KDD99 are selected by maximum information gain values and then the prior and conditional probabilities are used to construct a detection model using the selected attributes. In the second phase, the testing data of KDD99 passed through the trained model to detect the intrusions and find the detection rates and false positives of the detection model. In the experiment 41 attributes of KDD99 dataset are labeled in order as A_1, A_2, A_3, A_4, A_5, A_6, A_7, A_8, A_9, A_{10}, A_{11}, A_{12}, A_{13}, A_{14}, A_{15}, A_{16}, A_{17}, A_{18}, A_{19}, A_{20}, A_{21}, A_{22}, A_{23}, A_{24}, A_{25}, A_{26}, A_{27}, A_{28}, A_{29}, A_{30}, A_{31}, A_{32}, A_{33}, A_{34}, A_{35}, A_{36}, A_{37}, A_{38}, A_{39}, A_{40}, A_{41}. All experiments were performed using an Intel Core 2 Duo Processor 2.0 GHz processor (2 MB Cache, 800 MHz FSB) with 1 GB of RAM. We selected the important attributes from KDD99 dataset using our proposed algorithm and found out that 19 attributes are important and 22 attributes are redundant or less important. The 19 important attributes are A_1, A_3, A_4, A_5, A_6, A_8, A_9, A_{10}, A_{11}, A_{13}, A_{15}, A_{16}, A_{17}, A_{18}, A_{19}, A_{23}, A_{24}, A_{32}, and A_{33}. On the other side the 22 redundant attributes are A_2, A_7, A_{12}, A_{14}, A_{20}, A_{21}, A_{22}, A_{25}, A_{26}, A_{27}, A_{28}, A_{29}, A_{30}, A_{31}, A_{34}, A_{35}, A_{36}, A_{37}, A_{38}, A_{39}, A_{40}, and A_{41}. After identifying the important attributes from the KDD99 dataset we calculated prior and conditional probabilities to build the intrusion detection model. Then the testing dataset of KDD99 is used on detection model to classify the examples as an attack or normal. The performance comparison based on detection rate (DR) and false positives (FP) between 41 attributes and 19 attributes for 5 attack classes on KDD99 dataset using ID3 algorithm, naïve Bayesian classifier, and our proposed algorithm are listed in Table 3, Table 4, Table 5, and Table 6.

TABLE III.
DETECTION RATES (%) USING 41 ATTRIBUTES

Classes	ID3 Algorithm	NB Classifier	Proposed Algorithm
Normal	99.63	99.27	99.65
Probe	97.85	99.11	99.21
DOS	99.51	99.69	99.71
U2R	49.21	64.00	99.17
R2L	92.75	99.11	99.25

TABLE IV.
DETECTION RATES (%) USING 19 ATTRIBUTES

Classes	ID3 Algorithm	NB Classifier	Proposed Algorithm
Normal	99.71	99.65	99.82
Probe	98.22	99.35	99.72
DOS	99.63	99.71	99.75
U2R	86.11	64.84	99.47
R2L	97.79	99.15	99.35

TABLE V.
FALSE POSITIVES (%) USING 41 ATTRIBUTES

Classes	ID3 Algorithm	NB Classifier	Proposed Algorithm
Normal	0.10	0.08	0.07
Probe	0.55	0.45	0.42
DOS	0.04	0.04	0.04
U2R	0.14	0.14	0.12
R2L	10.03	8.02	7.87

TABLE VI.
FALSE POSITIVES (%) USING 19 ATTRIBUTES

Classes	ID3 Algorithm	NB Classifier	Proposed Algorithm
Normal	0.06	0.05	0.05
Probe	0.51	0.32	0.28
DOS	0.04	0.04	0.03
U2R	0.12	0.12	0.10
R2L	7.34	6.87	6.24

Therefore, it is clear from the above result that significant attribute selection improves the performance of detection model.

V. CONCLUSION

In this paper, we proposed a new learning approach for network intrusion detection that performs data reduction by selecting important subset of attributes. The performance of our proposed approach on the KDD99 benchmark intrusion detection dataset achieved balance detection rates for five classes. It also reduced the false positives compared to ID3 algorithm and naïve Bayesian classifier. The experimental results manifest that significant attribute selection improves the

performance of IDS. The attacks of KDD99 dataset detected with 99% accuracy using our proposed approach. The future work focus on improving the false positives of R2L attack and apply the detection model into real world IDS.

ACKNOWLEDGMENT

Support for this research received from the labratoire ERIC, Universite Lumiere Lyon 2 – 5av. Pierre Mendes – France – 69676 BRON Cedex.

REFERENCES

[1] Richard Heady, George Luger, Arthur Maccabe, and Mark Servilla, "The Architecture of a Network Level Intrusion Detection System," Technical report, University of New Mexico, 1990.

[2] James P. Anderson, "Computer Security Threat Monitoring and Surveillance," Technical report, James P. Anderson Co., Fort Washington, Pennsylvania. April 1980.

[3] Dorothy E. Denning, "An Intrusion Detection Model," IEEE Transaction on Software Engineering, SE-13(2), 1987, pp. 222-232.

[4] Mukkamala S., Sung A. H. and Abraham A., "Intrusion Detection using Ensemble of Soft Computing Paradigms," In Proceedings of the 3rd International Conference on Intelligent Systems Design and Applications, Springer Verlag Germany, 2003, pp. 209-217.

[5] W.K. Lee, and S.J.Stolfo, "A Data Mining Framework for Building Intrusion Detection Models," In Proceedings of the IEEE Symposium on Security and Privacy, Oakland, CA: IEEE computer Society Press, 1999, pp. 120-132.

[6] Commission of the European Communities, "Information Technology Security Evaluation Criteria," Version 2.1.1991.

[7] MIT Lincoln Laboratory, http://www.ll.mit.edu/IST/idaval/

[8] Marcus A. Maloof, and Ryszard S. Michalski, "Incremental learning with partial instance memory," In Proceedings of Foundations of Intelligent Systems: 13th International Symposium, ISMIS 2002, volume 2366 of Lecture Notes in Artificial Intelligence, Springer-Verlag, 2002, pp. 16-27.

[9] Wenke Lee, "A Data Mining Framework for Constructing Features and Models for Intrusion Detection Systems," PhD thesis, Columbia University, 1999.

[10] Wei Fan, "Cost-Sensitive, Scalable and Adaptive Learning using Ensemble-based Methods," PhD thesis, Columbia University, 2001.

[11] M.A. Maloof and R.S. Michalski, "A partial memory incremental learning methodology and its applications to computer intrusion detection," Reports of the Machine Learning and Inference Laboratory MLI 95-2, Machine Learning and Inference Laboratory, George Mason University, 1995.

[12] Kenneth A. Kaufman, Guido Cervone, and Ryszard S. Michalski, "An application of Symbolic Learning to Intrusion Detection: Preliminary Result from the LUS Methodology," Reports of the Machine Learning and Inference Laboratory MLI 03-2, Machine Learning and Inference Laboratory, George Mason University, 2003.

[13] C. Elkan. (2007, Jan, 27). Results of the KDD'99 Knowledge Discovery Contest [Online]. Available: http://www-cse.ucsd.edu/users/elkan/clresults.html

[14] Tadeusz Pietraszek, and Chris Vanden Berghe, "Defending Against Injection Attacks through Context-sensitive String Evaluation," In Recent Advances in Intrusion Detection (RAID2005), volume 3858 of Lecture Notes in Computer Science, Seattle, WA, 2005, Springer-Verlag, pp. 124-145.

[15] The PHP Group, PHP hypertext preprocessor, Web page at http://www.php.net. 2001-2004

[16] The phpBB group, phpBB,com, Web page at http://www.phpbb,com. 2001-1004

[17] Martin Roesch, "SNORT: The Open Source Network Intrusion System," Official web page of Snort at http://www.snort.org, 1998-2005.

[18] X. Xu, X.N. Wang, "Adaptive network intrusion detection method based on PCA and support vector machines," Lecture Notes in Artificial Intelligence, ADMA 2005, LNAI 3584, 2005, pp. 696-703.

[19] D.Y. Yeung, and Y.X. Ding, "Host-based intrusion detection using dynamic and static behavioral model," Pattern Recognition, 36, 2003, pp. 229-243.