ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.18, No.10, October 2007, pp.2635−2644
DOI: 10.1360/jos182635
© 2007 by *Journal of Software*. All rights reserved.
E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

一种用于网络取证分析的模糊决策树推理方法*

刘在强+, 林东岱, 冯登国

(中国科学院 软件研究所 信息安全国家重点实验室,北京 100080)

Fuzzy Decision Tree Based Inference Techniques for Network Forensic Analysis

LIU Zai-Qiang+, LIN Dong-Dai, FENG Deng-Guo

(State Key Laboratory of Information Security, Institute of Software, The Chinese Academy of Sciences, Beijing 100080, China)

+ Corresponding author: Phn: +86-10-62528254 ext 801, E-mail: liuzq@is.iscas.ac.cn

Liu ZQ, Lin DD, Feng DG. Fuzzy decision tree based inference techniques for network forensic analysis. *Journal of Software*, 2007,18(10):2635−2644. http://www.jos.org.cn/1000-9825/18/2635.htm

Abstract: Network forensics is an important extension to present security infrastructure, and is becoming the research focus of forensic investigators and network security researchers. However many challenges still exist in conducting network forensics: The sheer amount of data generated by the network; the comprehensibility of evidences extracted from collected data; the efficiency of evidence analysis methods, etc. Against above challenges, by taking the advantage of both the great learning capability and the comprehensibility of the analyzed results of decision tree technology and fuzzy logic, the researcher develops a fuzzy decision tree based network forensics system to aid an investigator in analyzing computer crime in network environments and automatically extract digital evidence. At the end of the paper, the experimental comparison results between our proposed method and other popular methods are presented. Experimental results show that the system can classify most kinds of events (91.16% correct classification rate on average), provide analyzed and comprehensible information for a forensic expert and automate or semi-automate the process of forensic analysis.

Key words: network forensics; fuzzy decision tree; data-mining; feature extraction; intrusion detection

摘 要: 网络取证是对现有网络安全体系的必要扩展,已日益成为研究的重点.但目前在进行网络取证时仍存在很多挑战:如网络产生的海量数据;从已收集数据中提取的证据的可理解性;证据分析方法的有效性等.针对上述问题,利用模糊决策树技术强大的学习能力及其分析结果的易理解性,开发了一种基于模糊决策树的网络取证分析系统,以协助网络取证人员在网络环境下对计算机犯罪事件进行取证分析.给出了该方法的实验结果以及与现有方法的对照分析结果.实验结果表明,该系统可以对大多数网络事件进行识别(平均正确分类率为 91.16%),能为网络取证人员提供可理解的信息,协助取证人员进行快速高效的证据分析.

关键词: 网络取证;模糊决策树;数据挖掘;特征提取;入侵检测

中图法分类号: TP393 **文献标识码**: A

* Supported by the National High-Tech Research and Development Plan of China under Grant Nos.2006AA01Z412, 2006AA01Z437, 2006AA01Z433 (国家高技术研究发展计划(863))

Received 2006-01-16; Accepted 2006-03-09

© 中国科学院软件研究所 http://www.jos.org.cn

1 Introduction

With the fast development and growth in networking connectivity, complexity and activity, there has been an increase in the number of crimes committed within networks. This is forcing both enterprises and law enforcement to undertake highly specialized investigations. Network forensics is the act of capturing, recording and analyzing network audit trails in order to discover the source of security breaches or other information assurance problems[1]. The biggest challenge in conducting network forensics is the sheer amount of data generated by the network. Besides this, the comprehensibility of the process of analyzing evidences that are extracted from collected data is also an important aspect for forensic experts. Therefore, the investigators need the aid of an effective, comprehensible and automated analyzing system for network intrusion forensics. In this paper, we propose a fuzzy decision tree based system for network intrusion forensics that can detect and analyze efficiently computer crime in networked environments, and locate digital evidences automatically.

The remainder of the paper is organized as follows: Section 2 discusses the related work such as network forensics and fuzzy decision tree system. Section 3 describes the proposed Fuzzy Decision Tree based system for network forensics. Section 4 explains the experimental data which is used in this paper and shows the experimental results. Finally, a discussion of conclusion and further issues in network forensics are given in Section 5.

2 Related Work

2.1 Network forensics

The term network forensics was introduced by the computer security expert Marcus Ranum in the early 90's[2], and is borrowed from the legal and criminology field where "forensics" pertains to the investigation of crimes. Network forensic systems are designed to identify unauthorized use, misuse, and attacks on information. Usually, network forensics which is based on audit trails is difficult and time-consuming process. Recently artificial intelligence technologies, such as artificial neural network (ANN) and support vector machine (SVM)[1], were developed to extract significant features for network forensics to automate and simplify the process. These techniques are effective in reducing the computing-time and increasing the intrusion detection accuracy to a certain extent, but they are limited in forensic analysis. Particularly, these systems are complex, and the results produced by these methods lack enough comprehensibility. Besides these, an evidence graph-based analysis method has been proposed[3], and although it is nice to present evidence correlation in graphic mode, this system is still a prototype and lacks the effective capability of inference. Finally, a fuzzy expert system has also been proposed for network forensics[4], but it still asks for experts to build a knowledge base and it lacks the capability of self-learning. The fuzzy decision tree-based forensic system proposed in this paper can effectively solve the above problems while keeping better analytical result.

2.2 Fuzzy decision tree

Decision trees were popularized by Quinlan with the ID3 program[5]. ID3 is based on the Concept Learning System algorithm. ID3 works by searching through the attributes of the training instances $\{E|e_1,e_2,\ldots,e_i,\ldots,e_N\}$ (where N=number of possible training samples) and extracting the attribute from attribute set $\{A|a_1,a_2,\ldots,a_j,\ldots,a_M\}$ (where M=number of possible values of an attribute) that best separates the given examples. The algorithm uses a greedy search to choose the best attribute and never looks back to reconsider earlier choices. We need to note that ID3 algorithm usually work well in symbolic domains, but does not work in a numerical decision. An extension of ID3 is the C4.5 and C5.0 algorithms, which extend the domain of classification from categorical attributes to

© 中国科学院软件研究所 http://www.jos.org.cn

numeric ones. Although decision tree technologies have already been shown to be interpretable, efficient, problem independent and able to treat large scale applications, they are also recognized as highly unstable classifiers with respect to minor perturbations in the training data, in other words, methods presenting high variance. Fuzzy logic brings in an improved in these aspects due to the elasticity of fuzzy set formalism. Fuzzy sets and fuzzy logic allow the modeling of language-related uncertainties, while providing a symbolic framework for knowledge comprehensibility[6]. Up to date, many algorithms have merged fuzzy representation, with its approximate reasoning capabilities, and symbolic decision trees while preserving advantages of both: uncertainty handling and gradual processing of the former with the comprehensibility, popularity, and ease of application of the latter[7,8]. It will further increase the representative power and applicability of decision trees by amending them with an additional knowledge component based on fuzzy representation.

3 Fuzzy Decision Tree-Based Network Forensic System

We develop a network forensic system based on fuzzy decision tree technology (NFSFDT). NFSFDT consists of the following components: Traffic Capturer, Feature Extractor, Forensic Analyzer. Figure 1 shows the architecture of the proposed system. The following sections detail the components respectively.

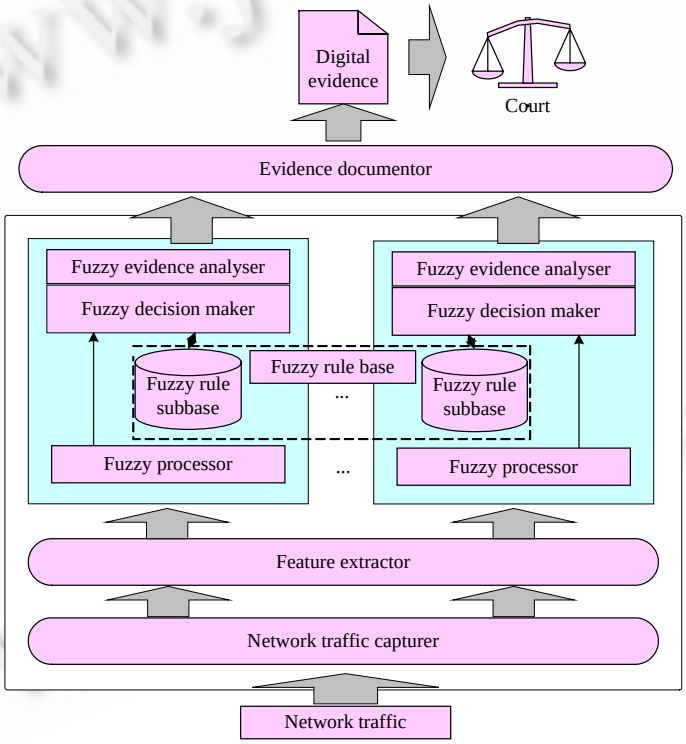

Fig.1 NFSFDT system

3.1 Traffic capturer

The Traffic Capturer component is responsible for network traffic capture and preparation for traffic analysis. The process of traffic capture is the first step of the proposed forensic system. While the capturing function is simple and straightforward, it provides the base information for other components of the forensic system. Currently the traffic capturer is based on the well-known packet capture program—TcpDump[9].

© 中国科学院软件研究所 http://www.jos.org.cn

3.2 Feature extractor

Feature Extractor performs extracting features on the "network traffic" captured by Traffic Capturer component. Feature extraction and selection from the available data is important to the effectiveness of the methods employed. Under the network environment, there are many traffic features that can be used for intrusion detection or event analysis, such as, source address and port number, destination address and port number, timestamp, etc. For convenience, we use a group of features as a kind of data structure characterizing network traffic. The most popular data structure for network event analysis is the connection log that consists of source address and port features, destination address and port features, etc. It has many advantages: being readily available; much more compact in size than other log formats, such as packet logs; efficient due to not examining data stream contents; being identified as a unique connection. Even though connection records provide numerous features that are special to each connection, we still need some features to effectively analyze network events. Essential attributes provide vital information about connections, but we still need some of the secondary attributes, such as TCP flags, connection duration and the volume of data passed in each direction. The JAM Project found that combining temporal information with connection log significantly increased accuracy[10]. Usually temporal information is determined by calculating the average value of a feature (attribute), or by calculating the accumulated count of connections over a time window (such as Δt seconds) or n connections. The Feature Extractor extracts 41 different features in all consisting of connection logs and other calculating features. For more detail information about feature selection, please refer to Refs.[10,11].

3.3 Fuzzy evidence analyzer

The Fuzzy Evidence Analyzer component is the core component of NFSFDT including three sub-components: Fuzzy Preprocessor, Fuzzy Rule Bases, and Fuzzy Decision Maker. The following sections detail the above sub-components individually.

3.3.1 Fuzzy preprocessor

There exist two different kinds of domains for features extracted by the Feature Extractor: continuous and discrete (such as service type: tcp, udp, icmp). Each input variable's sharp (crisp) value needs to be first fuzzified into linguistic values before the Fuzzy Decision-maker processes them with the Rule Base. Unlike classical sets, a fuzzy set expresses the degree to which an element belongs to a set. The characteristic function of a fuzzy set is assigned to values between 0 and 1, which denotes the degree of membership of an element in a given set.

The Fuzzy Preprocessor uses two different ways to fuzzify the continuous and the discrete respectively. For the discrete features, the Fuzzy Preprocessor component uses the same technique as the classical set. For example, let $protocol\_type=\{tcp,udp,icmp\}$ be the set of protocol type, then the membership function of each protocol type can be expressed as follows

$$\mu_{type}(x)=discret(x)=\begin{cases}1, & x=protocal\ type\\ 0, & \text{otherwise}\end{cases},$$

where $x\in\{tcp,udp,icmp\}$, $type$ is a fuzzy set. Besides protocol_type feature, there are others discrete features (such as service type, flag, etc.), which use the same fuzzifying method.

For continuous features, we choose the trapezoidal function as their membership function. The trapezoidal set is very popular in fuzzy theory due to its computational and storage efficiency, and more important, it is interpretable and comprehensible. A trapezoidal membership function is specified by four parameters $\{A_n,B_n,C_n,D_n\}$ as follows

© 中国科学院软件研究所 http://www.jos.org.cn

$$\mu_n(x)=trapezoid(x,A_n,B_n,C_n,D_n)=\begin{cases}0, & \text{for } x\le A_n\\ (x-A_n)/(B_n-A_n), & \text{for } A_n<x\le B_n\\ 1, & \text{for } B_n<x\le C_n\\ (D_n-x)/(D_n-C_n), & \text{for } C_n<x\le D_n\\ 0, & \text{for } D_n<x\end{cases},$$

where $\mu_n(x)$ represents the membership function of the n-th fuzzy subset. Note that: if B_n=C_n in the above formula, then $\mu_n(x)$ will become a triangle membership function (see $\mu_2(x)$ in Figure 2). Fig.2 presents the fuzzy subsets of the universe of discourse num_failed_logins (a feature denoting the number of failed login attempts). Using membership functions defined for each fuzzy set of each linguistic variable, the degree of membership of a sharp feature value in each fuzzy set is determined.

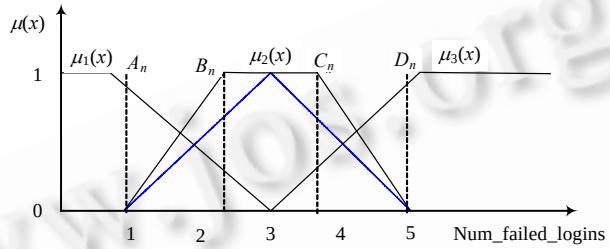

Fig.2 Membership function for num_failed_logins feature

Usually the membership function for continuous features can be user defined, but due to the large volume and the high dimensions of network data, it is very difficult to define the membership function for all the continuous features even for an expert. So NFSFDT uses an automatic approach to create the membership functions for each continuous feature. Assume a sample can be described by M attributes $\{A|a^{(1)},a^{(2)},\ldots,a^{(j)},\ldots,a^{(M)}\}$ and each attribute $a^{(j)}$ takes p_j values of a fuzzy subset $\{a_1^{(j)},a_2^{(j)},...,a_p^{(j)}\}$. The algorithm description for finding cut points and constructing member functions for continuous attributes as follows:

Step 1: If an attribute $a^{(j)}$ is continuous, then sort the training sample in ascending order according to the value of the attribute.

Step 2: Preprocess the values of the attribute $a^{(j)}$ in case the large value overwhelms the small one. For each attribute $a^{(j)}$ do the following condition calculation: If $\max(a_i^{(j)})/\min(a_i^{(j)})>\Gamma$, then $a_i^{(j)}=\log(a_i^{(j)})$, ($0<i\le N$). Note: here $a_i^{(j)}$ denotes the value of $a^{(j)}$ before being fuzzified; Γ denotes a positive integer, such as 10000.

Step 3: Search the candidate cut points of $a^{(j)}$. For each continuous attribute $a^{(j)}$ do the following condition calculation: If $a_i^{(j)}\in C_j$, $a_{i+1}^{(j)}\notin C_j$ and $a_i^{(j)}\ne a_{i+1}^{(j)}$, then $M_i=(a_i^{(j)}+a_{i+1}^{(j)})/2$ will be used as a candidate cut point. Note: There is no candidate cut point between adjoining data with equal attribute values and different classes[12].

Step 4: Calculate the membership functions of each continuous attribute ($a^{(j)}$). Calculation of the membership function is equal to the calculated values of the parameters $\{A_n,B_n,C_n,D_n\}$ (see Fig.2). The values and their ranges $\{A_n,B_n,C_n,D_n\}$ are described in Fig.3.

Note: $(1+\lambda)\bullet Median(\mu_{n-1})\le CP_n\le(1-\lambda)\bullet Median(\mu_n)$ and

$(1+\lambda)\bullet Median(\mu_n)\le CP_{n+1}\le(1-\lambda)\bullet Median(\mu_{n+1})$

where $0\le\lambda\le(1/2)$ and $Median(\mu_n)$ is the median of the set $a_i^{(j)}\,|\,CP_n\le a_i^{(j)}\le CP_{n+1}$.

© 中国科学院软件研究所 http://www.jos.org.cn

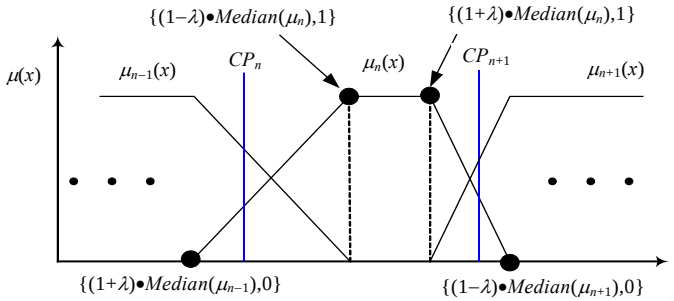

Fig.3 Calculating the membership functions of continuous attributes

In Step 3, we sometimes need to incorporate two or more adjoining cut points into a new one in case of getting too many fuzzy sets in step 4. Currently the system requires the intervention of a security expert to solve the problem.

3.3.2 Fuzzy rule bases

Fuzzy Rule Bases store the rules which are used by the Fuzzy Decision Makers to obtain a new fact. The process of building fuzzy rule bases is also the process of building fuzzy decision trees. For improving the efficiency of the decision-making and comprehensibility, we build an independent fuzzy subtree for each kind of service type respectively. The following is the process of building the fuzzy decision tree:

Step 1: Assume all the training samples $\{E|e_1,e_2,\ldots,e_i,\ldots,e_N\}$ (N denotes the count of training samples) with each sample based on its attributes as classified into C fuzzy subsets $\{\psi_1,\psi_2,\psi_3,\ldots,\psi_C\}$. Choose the service type attribute as the root node, and generate sub-tree $\{T|t_1,t_2,\ldots,t_K\}$ roots (K denotes the count of service type).

Step 2: Calculate the cut points and membership functions of $a^{(j)}$ using the method in the Fuzzy Preprocessor section under the sub-tree t_i.

Step 3: Calculate the fuzzy entropy of the chosen attribute $a^{(j)}$ for a subset $a_i^{(j)}$.

$$U_i^{(j)} = \sum_{m=1}^{C} -\rho_i^{(j)}(m)\log\rho_i^{(j)}(m),$$

where $\rho_i^{(j)}(m)$ denotes the relative frequency of $a_i^{(j)}$ (the ith subset of attribute j) with respect to ψ_m ($1\le m\le C$) and is defined as

$$\rho_i^{(j)}(m) = \frac{|a_i^{(j)} \cap \psi_m|}{|a_i^{(j)}|}.$$

And $|\bullet|$ denotes the cardinality of a fuzzy set.

Step 4: Choose the attribute t to split the instances at a given node.

$$t = attr\left\langle \min_{1\le j\le M}(IE^{(j)}) \right\rangle = attr\left\langle \min_{1\le j\le M} \sum_{i=1}^{p_j} \frac{|a_i^{(j)}|}{\sum_{i=1}^{p_j}|a_j^{(j)}|} U_i^{(j)} \right\rangle,$$

where $attr\langle\bullet\rangle$ is a function which returns the value of the $index(j)$ for which $IE^{(j)}$ (Information Entropy) is the smallest.

Step 5: Repeat Step 2 through Step 4 until:

- There are no more attributes for classification, or
- All data belongs to the same class, or
- The proportion of a data set of a class C_k is greater than or equal to a given threshold, or
- The number of elements in a data set is less than a given threshold

Step 6: Build next sub-tree t_{i+1} until all the K sub-trees are built.

After building the fuzzy decision tree, we can use the fuzzy decision tree to deduce the facts. The next section

© 中国科学院软件研究所 http://www.jos.org.cn

explains how to infer based on facts (test samples).

3.3.3 Fuzzy decision maker

The Fuzzy Decision Maker functions as a fuzzy inference engine. There are two main steps in the decision-making process of NFSFDT: Choose the fuzzy rule from the sub-base according to the service type; infer the type of network event based on the chosen fuzzy rule.

We use independent components to finish each of the two steps respectively. The Fuzzy Decision Maker consists of two units: Chooser and Inferencing-Maker. The Chooser receives the processed value by the Fuzzy Preprocessor component, and chooses the fuzzy rule sub-base in Fuzzy Rule Base according to the service type of the current network session. There are at lease two aspects of the benefits to categorizing the rule base according to service type: First, improving the efficiency of the process of decision-making by using the rule sub-base rather than the whole rule base to make decision; Second, improving the extensibility of the system.

The Inferencing-Maker is the inferencing component of the NFSFDT system. To decide the classification assigned to a test sample, we have to find “leaves” from the fuzzy decision tree (that is, the Fuzzy Rule Base) whose restrictions are satisfied by the sample, and combine their decisions into a single sharp response. Such decisions are very likely to cause conflicts. These conflicts are easy to find in the following conditions: In a single leaf (non-unique classifications of its examples), or across different leaves (different satisfied leaves have different examples, possibly with conflicting classifications)[7]. So we need to provide additional methods to guarantee true decision-making under these conditions. For simplicity the Inferencing-Maker uses the Maximum method. The decision function is defined as:

$$Class = \arg\left\langle \max_{1\le j\le C}\left\{ \sum_{set(Path)} \prod_{i\in Path} \mu\langle a^{(i)}\rangle \cdot \psi_j \right\} \right\rangle,$$

where $\mu\langle a^{(i)}\rangle$ denotes the value of membership degree for attribute $a^{(i)}$ of a test sample; $\arg\langle\bullet\rangle$ is a function which returns the value of the index (j) for which the product of $\mu\langle a^{(i)}\rangle$ is the maximum; Path is a matched path of a test sample (a fact) and $set(path)$ is the set of matched paths of the test sample.

For example, assume Fig.4 is a fuzzy decision tree and a test sample ts satisfies both path $P1$ and path $P2$, where $P1=\{a^{(1)},a^{(2)},a^{(5)},L1\}$ and $P2=\{a^{(1)},a^{(2)},a^{(3)},L2\}$ are paths from root node $a^{(1)}$ to leaves $L1$ and $L2$ respectively, ψ_1 and ψ_2 are target classes, and $set(Path)=\{P1,P2\}$, $C=2$.

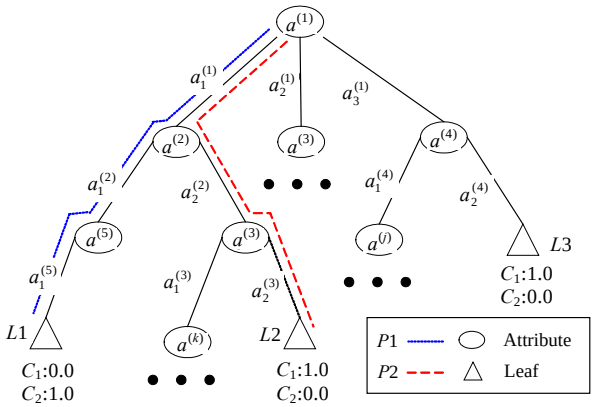

Fig.4 Fuzzy inference based on fuzzy decision tree

If the Decision-Maker judges that an attack or event occurred, then it will automatically notify the Evidence Documentor component (see Fig.1) to collect and document information from current network connection and

system logs.

Note: As for the details of the Evidence Documentor, we will discuss this in another paper.

4 Experiment and Result

The data for our experiments was prepared by the 1998 DARPA intrusion detection evaluation program from MIT Lincoln Labs[13]. The following experiment is based on the 10% train data subset with 494,021 data records. Each record has 41 attributes for each connection plus one class label. In order to make the results even more comprehensible, we categorize the target into five different classes {R2L, DOS, Probe U2R, Normal} rather than the usual two classes {Normal, Abnormal}. R2L denotes unauthorized access from a remote machine, such as guessing a password; DOS denotes denial-of-service, such as smurf attack; U2R denotes unauthorized access to local superuser privileges, such as various "buffer overflow" attacks; "Probe" denotes surveillance and other probing, such as host or port scanning[10].

There are several ways to evaluate the performance and efficiency of a classifier. Usually TP Rate (True Positive Rate) and FP Rate (False Positive Rate) are employed. In our experiment we also use RECALL and PRECISION measures to characterize the performance of the NFSFDT system

$$RECALL=TP,\quad PRECISION=\frac{TP}{TP+FP}.$$

Cross validation is a popular method of model evaluation, and in all our experiments we use 5-fold cross validation. The data set is divided into 5 subsets, and the following method is repeated *5* times. Each time, one of the 5 subsets is used as the test set and the other 4 subsets are put together to form a training set. Then the average error across all 5 trials is computed. Table 1 shows the result of the NFSFDT using the dataset with different measures. From Table 1, we can see that the NFSFDT has good performance, and a correctly classified instance rate that reaches 91.16% on average. But the system failed to classify "spy attack" which belongs to R2L type due to the lack of training samples (only 2 spy samples exists in the dataset). The experiment results also prove that the performance of the NFSFDT still depends on the quality of train samples to some degree just like other classifiers.

Table 1 Experimental result (λ=0.25)

Class	TP rate	FP rate	PRECISION	RECALL
R2L	0.901	0.064	0.861	0.901
DOS	0.925	0.014	0.899	0.925
Probe	0.956	0.03	0.943	0.956
U2R	0.784	0.042	0.715	0.840
Normal	0.936	0.01	0.978	0.936

In order to verify the performance, we employ some popular data mining algorithms (Naive Bayes algorithm[14], SMO algorithm[15], Decision Table majority classifier(DT)[16], C4.5[17]) to do the comparison experiments which using the Weka tool (Weka is an open source data mining software package[18]). In Weka, the above four algorithms are implemented by weka.classifiers.bayes.NaiveBayes class, weka.classifiers.functions.SMO class, weka.classifiers.rules.DecisionTable class, and weka.classifiers.trees.J48 class respectively.

The experimental results are illustrated in Figs.5 and 6. Figure 5 shows the results comparing the TP rate, while Fig.6 showing the corresponding results of PRECISION measurements (FDT denotes the algorithm proposed in this paper). From Figs.5 and 6, we can see the method proposed in this paper is comparable with the SMO algorithm and better than other algorithms in TP and PRECISION measurements. Besides this, it also shows that all the algorithms have better performances in detecting DOS and Probe attack types than U2L, which is the result of different natural distribution of training data. Even though the SMO algorithm presents a good performance in detecting different attacks, the SMO algorithm is based on a kernel function (RBF kernel is used in the experiment), so it lacks

deserved comprehensibility for forensic analysis. Therefore we can conclude from the results that the method proposed in this paper (FDT) has the potential of being most useful for network forensic analysis.

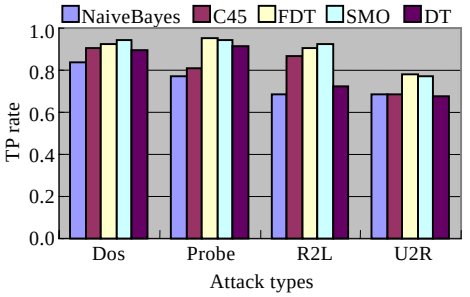

Fig.5 Comparing TP rate-attack type

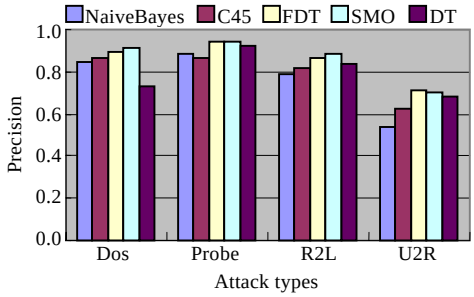

Fig.6 Comparing precision-attack type

5 Conclusion

In this paper, we developed an automated network intrusion forensic system (NFSFDT), which can produce interpretable and accurate results for forensic experts by applying a fuzzy logic based decision tree data mining system. The main characteristics of the NFSFDT consist of three aspects: making the output results of the NFSFDT easier to understand by using the fuzzy decision tree technology; improving the efficiency of the forensic analysis using automatic fuzzy inference; making the system parallel by building sub-trees based on network service type. The proposed method overmatches the existing methods[1,3,4] at least in one aspect. The experiments also proved that the system has the potential for automating the process of network forensic analysis.

Our future work plan is to develop and implement a parallel network forensic analysis algorithm based on fuzzy decision trees which can build global and even more accurate classifiers with distributed data sets for forensic investigators.

References:

[1] Mukkamala S, Sung AH. Identifying significant features for network forensic analysis using artificial intelligent techniques. Int'l Journal of Digital Evidence, 2003,1(4):1−7.

[2] Ranum M. Network flight recorder. http://www.ranum.com/

[3] Wang W, Daniels TE. Network forensics analysis with evidence graph. In: Proc. of the 2005 Digital Forensic Research Workshop (DFRWS). New Orleans, 2005.

[4] Kim JS, Kim M, Noh BN. A fuzzy expert system for network forensics. In: Proc. of the ICCSA 2004. LNCS 3043, 2004. 175−182.

[5] Quinlan JR. Induction on decision trees. Machine Learning, 1986,1(1):81−106

[6] Zadeh LA. Fuzzy logic and approximate reasoning. Synthese, 1975,30:407−428.

[7] Janikow CZ. Fuzzy decision trees: Issues and methods. IEEE Trans. on Systems, Man and Cybernetics, 1998,28(1):1−14.

[8] Olaru C, Wehenkel L. A complete fuzzy decision tree technique. Fuzzy Sets and Systems, 2003,138(2):21−254.

[9] http://www.tcpdump.org

[10] Stolfo SJ, Fan W, Lee W. Cost-Based modeling and evaluation for data mining with application to fraud and intrusion detection: Results from the JAM project. In: Proc. of the DARPA Information Survivability Conf. 2000.

[11] Lee W, Stolfo SJ. Data mining approaches for intrusion detection. In: Proc. of the 7th USENIX Security Symp. San Antonio, 1998.

[12] Zeidler J, Schlosser M. Continuous valued attributes in fuzzy decision trees. In: Proc. of the IPMU'96. 1996. 395−400.

[13] http://www.ll.mit.edu/IST/ideval/index.html

[14] Kohavi R. Scaling up the accuracy of naive-Bayes classifiers: A decision tree hybrid. In: Proc. of the 2nd Int'l Conf. on Knowledge Discovery and Data Mining. 1996.

[15] Keerthi SS, Shevade SK, Bhattacharyya C, Murthy KRK. Improvements to Platt's SMO algorithm for SVM classifier design. Neural Computation, 2001,13(3):637−649.

[16] Kohavi R. The power of decision tables. In: Proc. of the European Conf. on Machine Learning. 1995.

[17] Quinlan R. C4.5: Programs for Machine Learning. San Mateo: Morgan Kaufmann Publishers, 1993.

[18] http://www.cs.waikato.ac.nz/~ml/

LIU Zai-Qiang was born in 1976. He is a Ph.D. candidate at the Institute of Software, the Chinese Academy of Sciences. His current research areas are forensic computing and information security.

LIN Dong-Dai was born in 1964. He is a professor and doctoral supervisor at the Institute of Software, the Chinese Academy of Sciences. His research area is information security.

FENG Deng-Guo was born in 1965. He is a professor and doctoral supervisor at the Institute of Software, the Chinese Academy of Sciences, and a CCF senior member. His research areas are information security and network security.

Seventeenth International World Wide Web Conference (WWW2008) April 21-25, 2008

The International World Wide Web Conferences Steering Committee (IW3C2) cordially invites you to participate in the 17th International World Wide Web Conference (WWW2008), to be held on April 21-25, 2008 in Beijing, China. The conference series has become the premier venue for academics and industry to present, demonstrate, and discuss the latest ideas about the Web. The technical program for the five-day conference will include refereed paper presentations, plenary sessions, panels, and poster sessions. The WWW2008 program will also include Tutorials and Workshops, a W3C track, a Developers track, a WWW in China track, and Exhibitions.

IMPORTANT DATES

Submission Deadlines:

Workshop Proposals: October 1, 2007

Refereed Papers: November 1, 2007 (*HARD deadline; no extensions will be granted*)

Tutorial Proposals: November 1, 2007

Posters: January 25, 2008 (estimated)

Acceptance Notification: Refereed Papers – January 15, 2008 (tentative)

Conference dates: April 21 – 25, 2008

REFEREED PAPERS

WWW2008 seeks original papers describing research in all areas of the Web. Topics include but are not limited to: Browsers and User Interfaces; Data Mining; Industrial Practice and Experience; Internet Monetization; Mobility; Performance and Scalability; Rich Media Search; Search Security and Privacy; Semantic Web; Social Networks and Web 2.0; Technology for Developing Regions; Web Engineering; XML and Web Data.

General queries regarding WWW2008 submissions can be sent to ***submissions@www2008.org***. Other inquiries about the conference can be sent to ***info@www2008.org***.

The full call for papers, including detailed information about the scope of each track, formatting and submission requirements will be available soon from http://www2008.org.