A Fuzzy Data Mining Based Intrusion Detection Model *

Hai Jin, Jianhua Sun, Hao Chen, Zongfen Han
Cluster and Grid Computing Lab
Huazhong University of Science and Technology, Wuhan, 430074, China
{hjin, jhsun, haochen, zfhan}@hust.edu.cn

Abstract

With the information increases explosively, data mining techniques are frequently employed to identify trends in the warehouse that may not be readily apparent. In this paper we apply fuzzy data mining techniques to security system and build a fuzzy data mining based intrusion detection model. Through normalizing the data set and building fuzzy similar matrix of the network connections in the data set, network connections are clustered into different classes.

1 Introduction

Intrusion detection is a critical component of secure information systems. There are two main intrusion detection systems. *Anomaly intrusion detection system* (AIDS) such as IDES 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* (MIDS), which collects attack signatures, compares a behavior with these attack signatures, and signals intrusion when there is a match.

Many different approaches and techniques, such as fuzzy logic and neural networks, have been applied to anomaly intrusion detection. [6] generates fuzzy association rules from new audit data to detect whether an intrusion occurs or not. In [5], the *fuzzy intrusion recognition engine* (FIRE) uses fuzzy logic to assess whether malicious activity is taking place on a network. Bridges *et al*. apply fuzzy data mining techniques to the anomaly-based components [1]. [24] gives us a comparison of anomaly detection techniques and draws a conclusion that attentions should be paid to consider what are the most effective data streams to monitor. In order to build an efficient intrusion detection system, [22] is based on the techniques of SVMs and neural networks to identify important and useless input features.

*This paper is supported by Wuhan Hi-Tech Project under grant No. 20031003027

Intrusion detection systems usually pay their attention to detect attacks and intrusion and ignore the importance of redundant or repeated alarms that respond to the same occurrence of an attack. As a result, there are a lot of alarms and it is difficult to take appropriate actions. [4] provides methods to process alerts. The clustering and merging methods recognize alerts from the same intrusion and create only one alert totally to present these various alerts. A correlator in [17] is proposed to correlate related alerts and uncover the attack strategies behind sequences of attacks, based on the prerequisite and the consequence of each type of attacks.

In section 2, we discuss the related works. Section 3 introduces a set of relevant fuzzy cluster formulas, and describes how to calculate relationship between records. In section 4, we evaluate our intrusion detection model through experiments. Section 5 gives us a conclusion.

2 Related Works

Most intrusions occur via network using the network protocols to attack their targets. For example, during a certain intrusion, a hacker follows fixed steps to achieve his intention, first sets up a connection between a source IP address to a target IP, and sends data to attack the target. These kinds of connections are labeled attack connections and the rest connections are normal connection [8]. Generally, there are four categories of attacks [15]. They are:

- DOS (denial-of-service), for example, ping-of-death, syn flood, etc.
- PROBING, surveillance and probing, for example, port-scan, ping-sweep, etc.
- R2L, unauthorized access from a remote machine, for example, guessing password.
- U2R, unauthorized access to local superuser privileges by a local unprivileged user, for example, various buffer overflow attacks.

DOS and PROBING attacks involve many connections to some hosts in a very short period of time. R2L and U2R attacks are embedded in the data portions of packets, and normally involve only a single connection. Attack connec-

Proceedings of the 10th IEEE International Workshop on Future Trends of Distributed Computing Systems (FTDCS'04)
0-7695-2118-5/04 $20.00 © 2004 IEEE

tions and normal connections have their special feature values and flags in the connection head, and package contents can be used as signatures for normal determination and intrusion detection. Intrusions belong to the same intrusion category have identical or similar attack principles and intrusion techniques. Therefore they have identical or similar attack connections and are significantly different from normal connections.

3 Fuzzy Clustering

The aim of cluster analysis is the classification of network connections, or objects, according to similarities among them, and organizing objects into groups. A cluster is a group of objects that are more similar to other ones than to other clusters. Similarity is often defined by means of distance based upon the length from a data vector to some prototypical object of the cluster.

Since clusters can formally be seen as subsets of the data set, one possible classification method can be whether the subsets are fuzzy or crisp (hard). Hard clustering methods are based on classical set theory, and it requires an object that either does or does not belong to a cluster. *Fuzzy clustering methods* (FCM) allow objects to belong several clusters simultaneously with different degrees of membership. The data set, *U*, is thus partitioned into *r* fuzzy subsets. In many real situations, fuzzy clustering is more natural than hard clustering, as objects on the boundaries between several classes are not forced to fully belong to one of the classes. However, they rather are assigned to membership degrees between 0 and 1 indicating their partial memberships [10].

The data are typically observations of some phenomenon. Each object consists of m measured variables, grouped into an m-dimensional column vector $x_i = \{x_{i1}, x_{i2}, ..., x_{im}\}$. A set of n objects is denoted by $U = \{x_1, x_2, ...x_n\}$ and represented as a $n \times m$ matrix.

$$\begin{bmatrix} x_{11} & x_{12} & ... & x_{1m} \\ x_{21} & x_{22} & ... & x_{2m} \\ ... & ... & ... & ... \\ x_{n1} & x_{n2} & ... & x_{nm} \end{bmatrix} \quad (1)$$

3.1 Data Standardization

In order to remove the influence of dimension, we first standardize the data set. A collection of numeric data is standardized by subtracting a measure of central location (such as the mean or median) and divided by some measure of spread (such as the standard deviation, interquartile range or range). This yields data with a similarly shaped histogram with values centered around 0.

$$x'_{ik} = \frac{x_{ik} - \overline{x_k}}{s_k}, (i = 1, 2, ..., n, k = 1, 2, ..., m) \quad (2)$$

where $\overline{x_k}$ and s_k is the mean value and standard deviation of one feature or the kth dimension of U.

$$\overline{x_k} = \frac{1}{n} \sum_{i=1}^{n} x_{ik} \quad (3)$$

$$s_k = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (x_{ik} - \overline{x_k})^2} \quad (4)$$

Standardization transforms the mean of the set of feature values to zero, and the standard deviation to one. But x'_{ik} may not be in the interval [0,1]. After the following change, x''_{ik} is mapped into the interval [0,1].

$$x''_{ik} = \frac{x'_{ik} - \min_{1 \le i \le n} \{x'_{ik}\}}{\max_{1 \le i \le n} \{x'_{ik}\} - \min_{1 \le i \le n} \{x'_{ik}\}}, (k = 1, 2, ..., m) \quad (5)$$

3.2 Correlation Coefficient

In order to cluster objects, we build a fuzzy similar matrix to determine the correlation coefficient between x_i and x_j, $r_{ij} = R(x_i, x_j)$. The correlation coefficient can be calculated as

$$r_{ij} = \frac{\sum_{k=1}^{m} |x_{ik} - \overline{x_i}| \, |x_{jk} - \overline{x_j}|}{\sqrt{\sum_{k=1}^{m} (x_{ik} - \overline{x_i})^2} \cdot \sqrt{\sum_{k=1}^{m} (x_{jk} - \overline{x_j})^2}} \quad (6)$$

$$\overline{x_i} = \frac{1}{m} \sum_{k=1}^{m} x_{ik}, \overline{x_j} = \frac{1}{m} \sum_{k=1}^{m} x_{jk} \quad (7)$$

When $r_{ij} = -1$ there is a strong negative correlation between x_i and x_j, when $r_{ij} = 1$ there is a strong positive correlation, and when $r_{ij} = 0$ there is no correlation at all. If two objects are linearly dependent one of them is redundant; it is sufficient to select just one of them as an object.

Fuzzy similar matrix R is a fuzzy matrix and may not be transferable, or to say R is not a fuzzy equivalent matrix. In order to cluster these objects, we transfer R to an fuzzy equivalent matrix R^*. Through the square method in [25], we get the transitive closure of R, $t(R)$, which is the fuzzy equivalent matrix R^*. Based on R^*, by decreasing similarity threshold λ, a dynamic cluster result can be obtained [25].

Table 1. Sample Space

sample	features					
	1	2	...	k	...	m
x_1	x_{11}	x_{12}	...	x_{1k}	...	x_{1m}
x_2	x_{21}	x_{22}	...	x_{2k}	...	x_{2m}
...	...	...	...	...	...	...
x_i	x_{i1}	x_{i2}	...	x_{ik}	...	x_{im}
...	...	...	...	...	...	...
x_n	x_{n1}	x_{n2}	...	x_{nk}	...	x_{nm}
$\overline{x}$	$(\overline{x_1}$	$\overline{x_2}$	...	$\overline{x_k}$	...	$\overline{x_m})$

3.3 Determine Best Threshold λ

In the analysis of fuzzy cluster, the variance of $\lambda \in [0,1]$ causes different class results and forms a dynamic cluster result. This dynamic cluster result helps us understanding the relationship between objects. But it is usually required that λ be fixed and objects be classified to a concrete group.

$U = \{x_1, x_2, ...x_n\}$ is the sample space, and each sample has m features, $x_i = \{x_{i1}, x_{i2}, ..., x_{im}\}(i = 1, 2, ..., m)$. The raw data matrix is listed in Table 1. In Table 1, $\overline{x_k} = \frac{1}{n}\sum_{i=1}^{n} x_{ik}(k = 1, 2, ..., m)$. $\overline{x}$ is the center vector of sample space U.

Assume there are r kinds of classes for certain λ, and samples size of the jth class is n_j. The samples in the jth class are recorded as $x_1^{(j)}, x_2^{(j)}, ..., x_{n_j}^{(j)}$, and the center vector of the jth class is $\overline{x^{(j)}} = \left(\overline{x_1^{(j)}}, \overline{x_2^{(j)}}, ..., \overline{x_{m_j}^{(j)}}\right)$. $\overline{x_k^{(j)}}$ is the mean value the kth feature.

$$\overline{x_k^{(j)}} = \frac{1}{n_j}\sum_{i=1}^{n_j} x_{ik}^{(j)}(k = 1, 2, ..., m) \tag{8}$$

The random variable

$$F = \frac{\sum_{j=1}^{r} n_j \left|\left|\overline{x^{(j)}} - \overline{x}\right|\right|^2 / (r-1)}{\sum_{j=1}^{r}\sum_{i=1}^{n_j} \left|\left|x_i^{(j)} - \overline{x^{(j)}}\right|\right|^2 / (n-r)} \tag{9}$$

is a F distribution with $(n-1, n-r)$ degrees of freedom. Here

$$\left|\left|\overline{x^{(j)}} - \overline{x}\right|\right| = \sqrt{\sum_{k=1}^{m}\left(\overline{x_k^{(j)}} - \overline{x_k}\right)^2} \tag{10}$$

is the distance between $\overline{x^{(j)}}$ and $\overline{x}$. $\left|\left|x_i^{(j)} - \overline{x^{(j)}}\right|\right|$ is the distance between sample $x_i^{(j)}$ and the center vector $\overline{x^{(j)}}$ in the jth class. Numerator of F distribution suggests the distances between different classes, and denominator of F distribution suggests the distances between samples in the same class. Hence the larger F, the longer the distances between different classes, and the better the cluster result. If $F > F_\alpha(r-1, n-r)(\alpha = 0.05)$, the difference between classes is notable from statistical inference analysis [25].

4 Experiment

In this experiment, we use the raw data used by the 1999 KDD intrusion detection contest [8]. This database includes a wide variety of intrusions simulated in a military network environment. Being part of this database, test data file named *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 labelled normal or a certain attack type.

Table 2 shows the composition of the test data. There are total 39 types of data, including 38 types of attack and one normal type. The first, fourth and seventh columns of Table 2 mean the types of data, and rest columns are the frequency and percentage of each type. For example, the frequency of apache2 attack in the raw data is 794, and the percentage is 0.3.

Each data instance of test data has 42 fields. In order to illustrate conveniently, we number the 42 fields of each record in the *correcred.gz* file. Field 1 represents a certain attack name, field 2 represents the protocol type, and so on. In these 42 fields, 35 fields are numeric valued features, and 7 fields are nominal valued features. Among the nominal valued features, fields such as attack name, protocol_type, service and flag are combined together to label one attack.

4.1 Features Selection

When we process volumes of data, it is necessary to reduce the large number of features to a smaller number of features. There are 42 fields in each data record and it is hard to determine which fields are useful or which fields are trivial. However it may be feasible to correlate feature using formula (6). Correlation coefficients between fields are calculated by software SPSS [SPSS] and listed in Table 3.

Due to space limit, correlation coefficients of only the last 14 field are showed in Table 3. From Table 3 we can see that correlation coefficient between filed 28 and 29 is -0.792, the correlation coefficient between 28 and 30 is 0.331, and so on.

If the correlation coefficient of field i and j, $R(i,j)$, is larger than 0.8, there is a strong correlation between filed i and j, and select one of them to represent these two fields. From the correlation coefficients result produced by SPSS,

Table 2. Types of Data with Their Frequencies and Percentages in Test Data

Attack	Frequency	Percen.	Attack	Frequency	Percen.	Attack	Frequency	Percen.
apache2	794	.3%	named	17	.0%	sendmail	17	.0%
back	1098	.4%	neptune	58001	18.6%	smurf	164091	52.8%
buffer_overflow	22	.0%	nmap	84	.0%	snmpgetattack	7741	2.5%
ftp_write	3	.0%	normal	60593	19.5%	snmpguess	2406	.8%
guess_passwd	4367	1.4%	perl	2	.0%	sqlattack	2	.0%
httptunnel	158	.1%	phf	2	.0%	teardrop	12	.0%
imap	1	.0%	pod	87	.0%	udpstorm	2	.0%
ipsweep	306	.1%	portsweep	354	.1%	warezmaster	1602	.5%
land	9	.0%	processtable	759	.2%	worm	2	.0%
loadmodule	2	.0%	ps	16	.0%	xlock	9	.0%
mailbomb	5000	1.6%	rootkit	13	.0%	xsnoop	4	.0%
mscan	1053	.3%	saint	736	.2%	xterm	13	.0%
multihop	18	.0%	satan	1633	.5%			

Table 3. Correlation Coefficients between Fields from 28 to 41

	#28	#29	#30	#31	#32	#33	#34	#35	#36	#37	#38	#39	#40	#41
#28	1	-.792	331	-.035	.118	-.753	-.778	.336	-.449	-.047	-.090	-.099	.982	.994
#29	-.792	1	-.422	.076	-.152	.930	.961	-.397	.542	.060	-.483	-.480	-.798	-.797
#30	.331	-.422	1	.088	.034	-.398	-.409	.742	-.221	-.014	.126	.127	.323	.334
#31	-.035	.076	.088	1	-.334	-.012	.020	.002	-.189	.135	-.036	-.039	-.045	-.035
#32	.118	-.152	.034	-.334	1	-.006	-.109	.018	.240	-.411	.069	.070	.121	.118
#33	-.753	.930	-.398	-.012	-.006	1	.973	-.443	.567	-.044	-.465	-.461	-.761	-.758
#34	-.778	.961	-.409	.020	-.109	.973	1	-.450	.572	.040	-.479	-.475	-.787	-.783
#35	.336	-.397	.742	.002	.018	-.443	-.450	1	-.230	-.006	.152	.155	.356	.338
#36	-.449	.542	-.221	-.189	.240	.567	.572	-.230	1	-.040	-.280	-.278	-.455	-.449
#37	-.047	.060	-.014	.135	-.411	-.044	.040	-.006	-.040	1	-.027	-.028	-.049	-.047
#38	-.090	-.483	.126	-.036	.069	-.465	-.479	.152	-.280	-.027	1	.989	-.091	-.088
#39	-.099	-.480	.127	-.039	.070	-.461	-.475	.155	-.278	-.028	.989	1	-.087	-.098
#40	.982	-.798	.323	-.045	.121	-.761	-.787	.356	-.455	-.049	-.091	-.087	1	.987
#41	.994	-.797	.334	-.035	.118	-.758	-.783	.338	-.449	-.047	-.088	-.098	.987	1

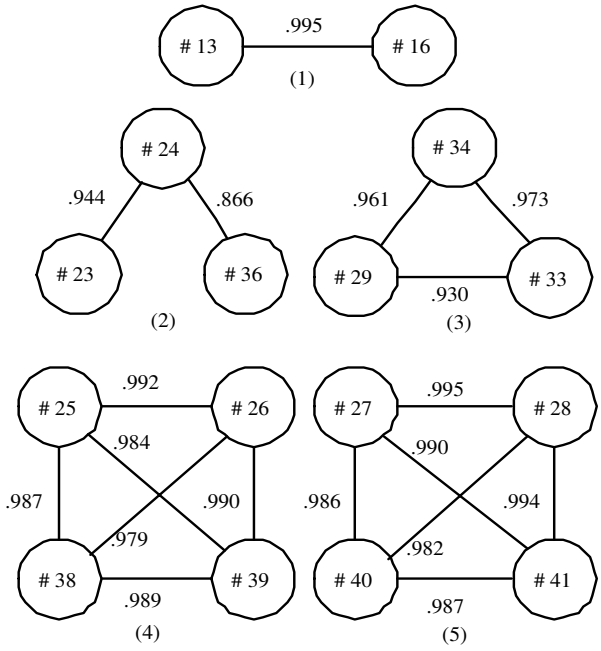

Figure 1. Graphic Representation of Correlation Coefficients

five groups in which elements correlate tightly are listed in the following.

- R(13,16)=0.995
- R(23,24)=0.944, R(24,26)=0.866
- R(29,33)=0.930, R(33,34)=0.973, R(29,34)=0.961
- R(25,26)=0.992, R(25,38)=0.987, R(25,39)=0.984, R(26,38)=0.979, R(26,39)=0.990, R(38,39)=0.989
- R(27,28)=0.995, R(27,40)=0.986, R(27,41)=0.990, R(28,40)=0.982, R(28,41)=0.994, R(40,41)=0.987

Figure 1 gives us the graphic representation of correlation coefficients of the above five groups. Figure 1 (4) is a complete graph, which means that correlation coefficient of each two fields is larger 0.8. We select one field among the four to represent these fields. In Figure 1, there are total 11 redundant fields, and these fields are omitted in the following process.

4.2 Classify Attack and Normal Records

We classify the test data into classes using the formulas in section 3. Table 4, 5, and 6 show the details of the cluster result. There are total 132 classes, some of which include only one type of attack or only normal records, like classes in Table 4, and rest of which include several types. In Table 4, the second line means that the 15th class contains the guess_passwd attack with protocol tcp, the attack target is pop_3 service, and the attack connection is at SF state. There are 3640 guess_passwd attack records in the 15th class, a percentage of 1.17. Table 5 shows what types of attacks are in the 50th class. The 50th class comprises 49 kind of neptune attacks with same protocol, connection state, and different services.

Classes in Table 4 and 5 comprise either attack records or normal records, so the number of false positive or false negative is zero. But the 20th class in Table 6 includes both attack and normal records. Number of normal records is 174, with a percentage of 0.06, so the false positive of this class is 0.0006.

If the percentage of the normal records in one class is larger than 50, this class is a normal class, and otherwise an attack class. Among these 132 classes, 34 classes are normal classes. 29 classes have only normal records and 5 normal classes include attack records. Among the 98 attack classes, 237 normal records are regarded as attack in 4 attack classes. From the result listed above, we can see that the performance of this model is excellent. The fuzzy clustering methods avoid the disadvantages caused by traditional data mining techniques. To avoid a hard definition between normal class and attack class, fuzzy clustering methods do not forced a behavior on the boundaries between several classes to fully belong to one of the classes.

5 Conclusions

In this paper we apply fuzzy data mining techniques to security system. Through normalizing the raw data and building fuzzy similar matrix, features selection, the raw data records are clustered into different classes.

Traditional data mining techniques are applied to security field to find intrusion patterns. However some behaviors are anomaly but not intrusion. We apply fuzzy clustering methods to intrusion detection to avoid a hard definition between normal class and certain intrusion class.

References

[1] S. M. Bridges and R. B. Vaughn. Fuzzy data mining and genetic algorithms applied to intrusion detection. *Proc. of the Twenty-third National Information Systems Security Conference*, pages 13–31, October 2000.

Table 4. Classes sample with Only One Type of Attack or Only Normal Records

class	Attack	Protocol	Service	Flag	Frequency	Percentage
15	guess_passwd	tcp	pop_3	SF	3640	1.17%
30	mailbomb	tcp	smtp	SF	5000	1.61%
67	normal	tcp	http	SF	39199	12.60%
78	normal	tcp	smtp	SF	3183	1.02%
84	normal	udp	domain_u	SF	3158	1.02%
86	normal	udp	private	SF	12796	4.11%
118	snmpgetattack	udp	private	SF	7733	2.49%

Table 5. The 50th Classes which Includes Several Types of Attack

Attack	Protocol	Service	Flag	Frequency	Attack	Protocol	Service	Flag	Frequency
neptune	tcp	auth	S0	15	neptune	tcp	login	S0	17
neptune	tcp	bgp	S0	18	neptune	tcp	mtp	S0	18
neptune	tcp	courier	S0	16	neptune	tcp	name	S0	18
neptune	tcp	csnet_ns	S0	15	neptune	tcp	netbios	S0	53
neptune	tcp	ctf	S0	24	neptune	tcp	netstat	S0	18
neptune	tcp	daytime	S0	14	neptune	tcp	nnsp	S0	16
neptune	tcp	discard	S0	14	neptune	tcp	nntp	S0	20
neptune	tcp	domain	S0	21	neptune	tcp	other	S0	14
neptune	tcp	echo	S0	21	neptune	tcp	pop_2	S0	18
neptune	tcp	efs	S0	15	neptune	tcp	pop_3	S0	11
neptune	tcp	exec	S0	17	neptune	tcp	printer	S0	17
neptune	tcp	ftp	S0	15	neptune	tcp	private	S0	16344
neptune	tcp	ftp_data	S0	33	neptune	tcp	remote_job	S0	20
neptune	tcp	gopher	S0	26	neptune	tcp	rje	S0	19
neptune	tcp	hostname	S0	16	neptune	tcp	shell	S0	15
neptune	tcp	http_443	S0	17	neptune	tcp	smtp	S0	18
neptune	tcp	imap4	S0	16	neptune	tcp	sql_net	S0	13
neptune	tcp	iso_tsap	S0	21	neptune	tcp	ssh	S0	15
neptune	tcp	klogin	S0	13	neptune	tcp	sunrpc	S0	11
neptune	tcp	kshell	S0	20	neptune	tcp	supdup	S0	12
neptune	tcp	ldap	S0	18	neptune	tcp	systat	S0	20
neptune	tcp	link	S0	22	neptune	tcp	time	S0	14
neptune	tcp	uucp_pat	S0	14	neptune	tcp	uucp	S0	16
neptune	tcp	whois	S0	20	neptune	tcp	vmnet	S0	13
neptune	tcp	Z39_50	S0	20					

Table 6. The 20th Classes which Includes both Attack and Normal Records

Attack	Protocol	Service	Flag	Frequency	Percentage
httptunnel	tcp	ftp_data	SF	1	0%
ipsweep	icmp	ecr_i	SF	6	0%
named	tcp	ftp_data	SF	4	0%
named	tcp	other	SF	2	0%
normal	icmp	ecr_i	SF	173	0.06%
normal	icmp	tim_i	SF	1	0%
pod	icmp	tim_i	SF	6	0%
portsweep	tcp	private	OT	4	0%
rootkit	tcp	ftp_data	SF	6	0%
smurf	icmp	ecr_i	SF	104089	33.46%
snmpguess	icmp	urp_i	SF	3	0%
warezmaster	tcp	ftp_data	S1	2	0%
xterm	tcp	ftp_data	SF	3	0%
httptunnel	tcp	ftp_data	SF	1	0%
Total				164302	33.52%

[2] G. C. Canavos. *Applied probability and statistical method*. Little, Brown and Company, Boston, 1984.

[3] G. Casella and R. L. Berger. *Statistical Inference*. Duxbury Press, Belmont, California, 1990.

[4] F. Cuppens and A. Miege. Alert correlation in a cooperative intrusion detection framework. *Proc. of the 2002 IEEE Symposium on Security and Privacy*, pages 202–216, 2002.

[5] J. E. Dickerson and J. A. Dickerson. Fuzzy network profiling for intrusion detection. *Proc. of 19th International Conference of the North American, Fuzzy Information Processing Society*, pages 301–306, 2000.

[6] G. Florez, S. M. Bridges, and R. B. Vaughn. An improved algorithm for fuzzy data mining for intrusion detection. *Proc. of Annual Meeting of the North American*, pages 457–462, 2002.

[7] S. A. Hofmeyr, S. Forrest, and A. Somayaji. Intrusion detection using sequences of system calls. *Journal of Computer Security*, 6(3):151–180, 1998.

[8] http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

[9] http://www.spss.come/.

[10] J. Jantzen. Neurofuzzy modelling. *Technical Report*, 1998.

[11] J. E. G. Jr. and J. W. Ulvila. Evaluation of intrusion detectors: a decision theory approach. *Proc. of IEEE Symposium on Security and Privacy*, pages 50–61, 2001.

[12] A. Kaufmann and M. M. Gupta. *Fuzzy mathematical models in engineering and management science*. Elsevier Science Inc., New York, 1998.

[13] W. Lee and S. Stolfo. Data mining approaches for intrusion detection. *Proc. of the Seventh USENIX Security Symposium*, January 1998.

[14] W. Lee, S. Stolfo, and P. Chan. Learning patterns from unix process execution traces for intrusion detection. *Proc. of AAAI Workshop: AI Approaches to Fraud Detection and Risk Management*, pages 50–56, July 1997.

[15] W. Lee, S. Stolfo, and K. Mok. A data mining framework for building intrusion detection models. *Proc. of the 1999 IEEE Symposium on Security and Privacy*, pages 120–132, May 1999.

[16] C. Li and G. Biswas. Conceptual clustering with numeric-and-nominal mixed data – a new similarity based system. *IEEE Trans. on Knowledge and Data Engineering*, pages 673–690, 1998.

[17] P. Ning and Y. Cui. An intrusion alert correlator based on prerequisites of intrusions. *Technical Report*, 2002.

[18] T. J. Ross. *Fuzzy logic with engineering application*. McGraw-Hill Companies, New York, 1995.

[19] S. L. Scott. A bayesian paradigm for designing intrusion detection systems. *Computational Statistics and Data Analysis*, 45(1):69–83, 2004.

[20] K. Sequeira and M. Zaki. Admit: anomaly-based data mining for intrusions. *Proc. of the eighth ACM SIGKDD international conference*, pages 386–395, 2002.

[21] J. Sun, H. Jin, H. Chen, and Z. Han. A data mining based intrusion detection model. *Proc. of Fourth International Conference on Intelligent Data Engineering and Automated Learning*, pages 677–684, 2003.

[22] A. H. Sung and S. Mukkamala. Identifying important features for intrusion detection using support vector machines and neural networks. *Proc. of Symposium on Applications and the Internet*, pages 209–217, 2003.

[23] T. Terano, K. Asai, and M. Sugeno. *Fuzzy systems theory and its applications*. Academic Press, Boston, 1992.

[24] C. Warrender, S. .Forrest, and B. Pearlmutter. Detecting intrusions using system calls: Alternative data models. *Proc. IEEE Symposium on Security and Privacy*, pages 133–145, 1999.

[25] J. J. Xie and C. P. Liu. *Fuzzy set theory and its applications*. Huazhong University of Science and Technology Press, China, 2000.