Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 2005 Proceedings

Americas Conference on Information Systems (AMCIS)

2005

Classifying Network Intrusions: A Comparison of Data Mining Methods

Louis W. Glorfeld
University of Arkansas, lglorfeld@cox-internet.com

Hilol Bala
University of Arkansas, hbala@walton.uark.edu

Robert Miller
University of Arkansas, rmiller@walton.uark.edu

Classifying Network Intrusions: A Comparison of Data Mining Methods

Louis W. Glorfeld
University of Arkansas
lglorfeld@cox-internet.com

Hillol Bala
University of Arkansas
hbala@walton.uark.edu

Robert Miller
University of Arkansas
rmiller@walton.uark.edu

ABSTRACT

Network intrusion is an increasingly serious problem experienced by many organizations. In this increasingly hostile environment, networks must be able to detect whether a connection attempt is legitimate or not. The ever-changing nature of these attacks makes them difficult to detect. One solution is to use various data mining methods to determine if the network is being attacked. This paper compares the performance of two data mining methods—i.e., a standard artificial neural network (ANN) and an ANN guided by genetic algorithm (GA)—in classifying network connections as normal or attack. Using connection data drawn from a simulated US Air Force local area network each method was used to construct a predictive model. The models were then applied to validation data and the results were compared. The ANN guided by GA (90.67% correct classification) outperformed the standard ANN (81.75% correct classification) significantly, indicating the superiority of GA-based ANN.

Key Words

Network intrusion detection, artificial neural network, genetic algorithm, classification methods.

INTRODUCTION

Information and communication technologies (ICTs) have become an integral part of organizations over the past few decades. Advanced electronic networks and complex information systems have enabled organizations to process, store and transmit digital data in order to increase productivity and organizational effectiveness. Organizations are now creating complex networked systems and opening their networks to external and internal stakeholders (Chin, 1999). Even though most users of these networks are legitimate, the number of illegitimate users who access and use these networks is increasing at an alarming rate (Zhu, Premkumar, Zhang and Chu, 2001). The increased interconnectivity of these complex networks, along with greater access and dependence on the Internet, has boosted the scale and scope of information technology security breaches and related crimes (Zhu et al., 2001; Cavusoglu, Mishra and Raghunathan, 2002). Therefore, network security has become a major concern for organizations and IT managers (Mehta and George, 2001).

Even with the emphasis placed on security today, the number of computer security breaches continues to increase each year. A 2002 survey conducted by the CSI/FBI reports that ninety percent (90%) of large corporations and agencies detected computer security breaches within the last twelve months, while eighty percent (80%) acknowledged financial losses due to security breaches (Power, 2002). According to a CERT/CC report, computer security vulnerabilities more than doubled in 2001, with 1,090 separate security holes reported in 2000, and 2,437 reported in 2001 (www.cert.org). The number of reported incidents also drastically increased with 21,756 documented in 2000 and 52,658 in 2002. Riptech has reported that general Internet attack trends are showing a sixty four percent (64%) annual growth rate based on continual 24 x 7 monitoring of Fortune 1000 companies (www.riptech.com).

While the costs of network security crime are difficult to measure (Newmann, 1999), experts believe that the costs are substantial and growing exponentially. Some researchers believe that these costs are doubling each year (Lukasik, 2000). The actual costs are almost impossible to identify because many cyber crimes are not reported. Ullman and Ferrera (1998) state that according to FBI estimates, only seventeen percent (17%) of computer crimes are reported to government authorities.

Estimates provided by the Computer Security Institute and the FBI show that organizations lost $124 million in 1999, $266 million in 2000, $378 million in 2001, and $456 million in 2002 due to various computer crimes (Power, 2002).

In recent years, network intrusion detection has become increasingly important (Zhu et al., 2001). While traditional approaches to network security have focused only on prevention, network intrusion detection can enable organizations to reduce undetected intrusion. In this study, we employ two data mining methods to classify network intrusions into multiple categories. Even though studies have compared the performance of various data mining methods, the results have been conflicting. Also, very few studies have examined the relative performance of various methods in the context of network intrusion detection. The absence of such studies is the motivation for the current work. The current work is different from earlier comparative studies on at least two counts. First, while most earlier works limit the number of classification categories to two groups, we use multiple groups for classification. Multiple group classification provides more insight into the types of network attacks and can guide organizations to place emphasis on specific attack types. Second, we use a an artificial neural network (ANN) guided by genetic algorithms (GA) which has been configured to maximize the classification rate as opposed to minimize the average error function normally used. To our knowledge, ours is one of the very few studies that employs such a technique.

The remainder of the paper proceeds as follows:

1. First, we provide the research background on network security and data mining methods, the two areas that form the foundation of this study.
2. Second, we introduce the data mining methods used in this study. These methods are standard ANN and an ANN guided by genetic algorithms.
3. Third, we discuss our research methodology that includes our data collection process, data preparation, and model development.
4. Finally, we present our results and discuss the implications, limitations, and future research directions.

BACKGROUND

Network Security

Lunt (1993) determined that there are two broad issues relevant to network security: *protection* and *detection*. More recently a third type of mechanism has been added: *response* (Cavusoglu, Raghunathan and Mishra, 2002). *Protection* techniques (also known as prevention) such as firewalls are designed to guard hardware, software, and user data against threats from both outsiders and malicious insiders (Zhu et al., 2001). *Detection* mechanisms like intrusion detection systems (IDS) try to detect the intrusions by collecting information from a variety of systems and network sources (e.g., operating systems or firewall log files) and then analyzing the information for signs of intrusion and misuse. Finally, *response* mechanisms deal with how a network responds to a security threat in order to protect the network from intrusion.

The major functions performed by an intrusion detection system, according to Zhu et al. (2001), are: (1) monitoring and analyzing user and system activity, (2) assessing the integrity of critical system and data files, (3) recognizing activity patterns reflecting known attacks, (4) responding automatically to detected activity, and (5) reporting the outcome of the detection process. Zhu et al. (2001) classified the intrusion detection tasks into two categories: misuse detection and anomaly detection. Misuse detection systems detect attacks based on well-known vulnerabilities and intrusions stored in a database, while anomaly detection systems detect deviations in activity from normal profiles.

Misuse detection systems use various techniques including rule-based expert systems, model-based reasoning systems, state transition analysis, genetic algorithms, fuzzy logic, and keystroke monitoring. The major limitation of misuse detection systems is their reliance on existing attack information and vulnerabilities. The rules and logic of the systems have to be continually updated as new forms of attacks are identified (Zhu et al., 2001). On the other hand, anomaly detection systems use techniques such as statistical analysis, sequence analysis, ANN, machine learning, and artificial immune systems (Zhu et al., 2001). These techniques reduce the reliance on updated rules by comparing the suspect access against “normal” usage patterns. In this paper, our primary focus is on anomaly detection.

OVERVIEW OF DATA MINING METHODS

As mentioned earlier, two data mining methods for classification are used in this study: a standard back propagation ANN and an ANN guided by GA. Standard ANNs have been used in several intrusion detection studies (e.g., Fox, Henning, Reed and Simonian, 1990; Debar, Becker and Siboni, 1992; Bonafacio, Cansian, Carvalho and Moreira, 1997; Lee, Stolfo and

Mok, 2000; Lippmann and Cunningham, 2000). However, to our knowledge, the ANN guided by GA used in this study, has never been used in the intrusion detection context. In this section, we provide a general description of ANN and GA.

Artificial Neural Networks (ANNs)

ANNs are the most dominant data mining method at present (Sung et al., 1999). Originally developed in an attempt to mimic the neural functions of a human brain, ANNs are powerful prediction and classification tools, and provide new opportunities for solving difficult problems that have been traditionally modeled using statistical approaches. Although many different multilayer ANN architectures are available for addressing classification problems, the backpropagation (BP) architecture is probably the most popular and widely used (Caudill, 1991; Zhu et al., 2001). A BP network consists of several components: (1) a set of neurons or processing units that receive and send signals from an outside environment or other neurons in the network using three layers – input, hidden, and output; (2) connectivity, which shows the interactivity between neurons; (3) activation/transfer functions, which convert the aggregated inputs into an output that is sent to other connected neurons; and (4) learning algorithms, which update the patterns and strength of the connectivity. Most of the time, the network starts with a random set of weights and adjusts the weights each time it detects an input-output pair of errors. During the training period, various classes of training data are fed into the network. In traditional BP network, training is an iterative process of minimizing the differences between the desired output and the actual output of the network using a least squares approach. The difference between the actual output and the desired output is calculated and back-propagated to the previous layer(s), which causes a series of adjustments to the function weights in order to reduce the observed output errors. The whole process is repeated layer by layer throughout the network. Based on the error signals received, connection weights are repeatedly updated until the network converges toward a stable state.

Genetic Algorithms (GA)

GAs are designed to solve problems in which an optimal solution is required given a very large number of possible solutions. Although mathematical techniques exist for these types of problems**,** the mathematical techniques do not work well when the number of possible solutions is very high. GAs handle these computationally intense problems by borrowing a technique from nature. Specifically, GAs use natural selection, or survival of the fittest, to find the optimal solution.

The process followed by a GA starts by generating a random population of possible solutions which are represented by binary strings known as chromosomes. The individual bits in each chromosome are called genes. Mimicking the nature, the genetic algorithm produces a solution based on an objective function (in our case, the maximum rate of correct classification) by evolving the chromosomes over a number of generations. In each generation the genetic algorithm performs three operations on the chromosomes: selection, crossover, and mutation. In selection, the genetic algorithm decides which chromosomes will survive into the next generation and which chromosomes will not—selecting the fittest members of the population. During crossover, pair of chromosomes swaps genes—mating the fittest members by combining the gene pool. Finally, in the mutation operation randomly selected genes have their values swapped. This randomness added in this operation prevents the GA from producing a sub-optimal solution. Thus, the use of these three operations by GAs allows them to handle a wide variety of problems while producing optimal solutions quickly.

RESEARCH METHODOLOGY

Data Collection

The dataset used in this study was initially acquired at the MIT Lincoln Labs. The data was collected as a part of the 1998 Intrusion Detection Evaluation Program funded by the Defense Advanced Research Projects Agency (DARPA). A standard set of data was audited including a wide variety of intrusions simulated in a military network environment. Lincoln Labs set up an environment to acquire nine weeks of raw TCP dump data for a local area network (LAN) simulating a typical U.S. Air Force LAN. They operated the LAN as it were a true Air Force environment. A version of this dataset was used in the 1999 KDD intrusion detection contest. We acquired the dataset from the KDD contest web site[1].

There are two datasets: training and validation. The training dataset had about five million connection records (4,898,430) and the validation dataset had about three million connection records (2,984,153). A connection is defined as a sequence of TCP packets starting and ending at some well defined times, between which data flows to and from a source IP address to a target IP address under some well defined protocol. In these datasets, each connection is labeled as either normal, or as an

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

attack, with exactly one specific attack type (see Table 1 for various attack types). The connection records have a set of features (or variables) which is shown in Table 2. As Table 2 shows, there are 41 input variables and 1 output variable.

Probe (surveillance and other probing)	**DOS** (denial-of-service)	**U2R** (unauthorized access to local superuser (root) privileges)	**R2L** (unauthorized access from a remote machine)
ipsweep	pod	perl	phf
portsweep	processtbale	ps	sendmail
saint	smurf	rootkit	snmpgetattack
satan	teardrop	sqlattack	snmpguess
nmap	udpstorm	sterm	worm
apache2	warezmaster	ftp_write	xlock
back	buffer_overflow	guess_passwd	xsnoop
land	httptunnel	httptunnel	warezclient
mailbomb	loadmodule	imap	warezmaster
neptune	multihop	multihop	

Table 1. Network Attack Types

Variable Name	**Data Type**	**Description**
duration	continuous.	length (number of seconds) of the connection
protocol_type:	symbolic.	type of the protocol, e.g. tcp, udp, etc.
service:	symbolic.	Network service on the destination, e.g., http, telnet, etc.
flag:	symbolic.	normal or error status of the connection
src_bytes:	continuous.	number of data bytes from source to destination
dst_bytes:	continuous.	number of data bytes from destination to source
land:	symbolic.	1 if connection is from/to the same host/port; 0 otherwise
wrong_fragment:	continuous.	number of ``wrong'' fragments
urgent:	continuous.	number of urgent packets
hot:	continuous.	number of ``hot'' indicators
num_failed_logins:	continuous.	number of failed login attempts
logged_in:	symbolic.	1 if successfully logged in; 0 otherwise
num_compromised:	continuous.	number of ``compromised'' conditions
root_shell:	continuous.	1 if root shell is obtained; 0 otherwise
su_attempted:	continuous.	1 if ``su root'' command attempted; 0 otherwise
num_root:	continuous.	number of ``root'' accesses
num_file_creations:	continuous.	number of file creation operations
num_shells:	continuous.	number of shell prompts
num_access_files:	continuous.	number of operations on access control files
num_outbound_cmds:	continuous.	number of outbound commands in an ftp session
is_host_login:	symbolic.	1 if the login belongs to the ``hot'' list; 0 otherwise
is_guest_login:	symbolic.	1 if the login is a ``guest''login; 0 otherwise
count:	continuous.	number of connections to the same host as the current connection in the past two seconds
srv_count:	continuous.	number of connections to the same service as the current connection in the past two seconds
serror_rate:	continuous.	% of connections that have ``SYN'' errors
srv_serror_rate:	continuous.	% of connections that have ``SYN'' errors
rerror_rate:	continuous.	% of connections that have ``REJ'' errors
srv_rerror_rate:	continuous.	% of connections that have ``REJ'' errors
same_srv_rate:	continuous.	% of connections to the same service

diff_srv_rate:	continuous.	% of connections to different services
srv_diff_host_rate:	continuous.	% of connections to different hosts
dst_host_count:	continuous.	number of connections to the same destination as the current connection in the past two seconds
dst_host_srv_count:	continuous.	number of connections to the same destination as the current connection in the past two seconds
dst_host_same_srv_rate:	continuous.	% of connections to the same destination
dst_host_diff_srv_rate:	continuous.	% of connections to different destination services
dst_host_same_src_port_rate:	continuous.	% of connections to different source ports
dst_host_srv_diff_host_rate:	continuous.	% of connections to different destination hosts
dst_host_serror_rate:	continuous.	% of connections that have ``SYN'' errors
dst_host_srv_serror_rate:	continuous.	% of connections that have ``SYN'' errors
dst_host_rerror_rate:	continuous.	% of connections that have ``REJ'' errors
dst_host_srv_rerror_rate:	continuous.	% of connections that have ``REJ'' errors
output	symbolic	Attack type for each connection record

Table 2. Variable List

It should be noted that the validation data has additional attack types not present in the training data. This makes the whole study more realistic. The training dataset contains a total of 24 attack types, while the validation dataset had 14 additional attack types.

Data Preprocessing

Data preprocessing is an important first step before doing any data mining experiments. In the preprocessing phase, we performed the following tasks: (1) we sampled the training and validation data to draw sets of a manageable size; (2) we coded the alphanumeric values into numeric values; and (3) we checked for missing values.

Sampling is performed using SAS Enterprise Miner's (version 4.3) sampling node. At first, we imported both the training and validation datasets into a SAS library. We generated two datasets based on random sampling each with 10,000 records for training and validation purposes. Sampling is recommended for extremely large datasets in order to optimize the model fitting time.

Coding was one of the most important tasks in our data preprocessing phase. Coding was required because the datasets contain alphanumeric data in four variables. We coded these alphanumeric data in such a way that the dataset only contained numeric data. The four variables that were coded are: protocol_type, service, flag, and classification. We followed a binary/dummy coding technique. After the coding, we ended up with 105 input variables and 5 output variables—i.e., normal, probe, DOS, U2R, and R2L (see Tables 1 and 2).

We investigated the datasets for missing data and could not find any missing data. We did not remove any influence points (outliers and leverage points) because in a real world network intrusion detection scenario, it is not practical to remove a connection record because that record may be a potential network intrusion.

Neural Network Development Methodology

Standard ANN

We used the SAS Enterprise Miner's (version 4.3) neural network feature as the standard ANN. SAS Enterprise Miner neural network is a standard multilayer BP network. The objective function was based on minimizing the average squared error. Objective function based on the maximization of correct classification rate was not available in Enterprise Miner. For the most part, we did not change the default configurations of the Enterprise Miner because we wanted it to be a representative of a standard ANN.

ANN Guided by GA

The basic architecture of the alternative method was very similar to that of the standard ANN. The primary difference was that in this method, the objective function was developed to maximize the correct classification rate and the network optimi-

zation methodology was based on the use of a GA as provided by the Gene Hunter DLL library and Microsoft Excel interface of the Ward Systems[2].

Maximization of the correct classification rate is not a well-behaved optimization problem and therefore, a GA is ideal for this optimization task. Before the genetic algorithm could be executed, the training and validation data had to be preprocessed beyond what was detailed earlier. To be specific, each data value had to be standardized between -1 and 1. This standardization was required because the equations used in the spreadsheet employed the hyperbolic tangent function. Since the function returns values from -1 to 1, all input values were standardized to the same range. In order to standardize the data, the minimum value and the range was calculated for each variable. With these values calculated, each data point's standardized value was calculated using the following function:

$$\text{standardized value} = ((\text{data point} - \text{minimum})/\text{range}) * 2 - 1$$

With the training and validation data standardized, the GA parameters were set to maximize the classification rate. Additional parameters that were set included the chromosome length which was set to continuous and the population size which was set to 1,000 chromosomes.

RESULTS

Table 3 presents the standard ANN classification for both training and validation data. For each attack type, the results show how many were classified correctly and how many were classified incorrectly. Using the training data, the standard ANN was able to produce a model with a fairly high correct classification rate (97.77%). When the model was applied to the validation data, the correct classification rate dropped significantly (81.75%). It should be noted that in both cases (training and validation), the neural network method was only able to classify normal connections and DOS attacks. It failed to correctly identify any of the other three attacks.

Training

	Predicted		
Actual	**Correct**	**Incorrect**	**Total**
Normal	1892	105	**1997**
Probe	0	87	**87**
DOS	7885	15	**7900**
U2R	0	5	**5**
R2L	0	11	**11**
Total	**9777**	**223**	**10000**

Correct Classification Rate = 97.77%

Validation

	Predicted		
Actual	**Correct**	**Incorrect**	**Total**
Normal	1086	898	**1984**
Probe	0	131	**131**
DOS	7089	331	**7420**
U2R	0	6	**6**
R2L	0	459	**459**
Total	**8175**	**1825**	**10000**

Correct Classification Rate = 81.75%

Table 3. Standard ANN Results

Table 4 presents the results of the ANN guided by GA. As the table shows, this method provides a greater degree of classification specificity. Using the training data, the genetic algorithm produced a model with a 99.07% correct classification rate. Applying the model to the validation data produced a correct classification rate of 90.67%. Not only did it report the number of correct and incorrect classifications by type, it also reported into which type the incorrectly classified connection records had been placed (see Table 5).

[2] This feature is no longer available in newer versions of Microsoft Excel. Therefore, we used an older version of Excel that came with Microsoft Office 95.

Training

	Predicted		
Actual	**Correct**	**Incorrect**	**Total**
Normal	1997	0	**1997**
Probe	24	63	**87**
DOS	7886	14	**7900**
U2R	0	5	**5**
R2L	0	11	**11**
Total	**9907**	**93**	**10000**

Correct Classification Rate = 99.07%

Validation

	Predicted		
Actual	**Correct**	**Incorrect**	**Total**
Normal	1942	42	**1984**
Probe	59	72	**131**
DOS	7066	354	**7420**
U2R	0	6	**6**
R2L	0	459	**459**
Total	**9067**	**933**	**10000**

Correct Classification Rate = 90.67%

Table 4. ANN guided by GA Results

Training

	Predicted					
Actual	**Normal**	**Probe**	**DOS**	**U2R**	**R2L**	**Total**
Normal	1997	0	0	0	0	**1997**
Probe	41	24	22	0	0	**87**
DOS	11	3	7886	0	0	**7900**
U2R	5	0	0	0	0	**5**
R2L	11	0	0	0	0	**11**
Total	**2065**	**27**	**7908**	**0**	**0**	**10000**

Correct Classification Rate = 99.07%

Validation

	Predicted					
Actual	**Normal**	**Probe**	**DOS**	**U2R**	**R2L**	**Total**
Normal	1942	1	41	0	0	**1984**
Probe	28	59	44	0	0	**131**
DOS	310	44	7066	0	0	**7420**
U2R	5	1	0	0	0	**6**
R2L	452	0	7	0	0	**459**
Total	**2737**	**105**	**7158**	**0**	**0**	**10000**

Correct Classification Rate = 90.67%

Table 5. Detailed Results for ANN guided by GA

DISCUSSION AND IMPLICATIONS

Given the aforementioned classification results, identifying the "best" data mining method by comparing correct classification rate was a straightforward exercise. The ANN guided by GA performed significantly better than the standard ANN. As mentioned earlier, the standard ANN returned a correct/incorrect classification by type. The ANN guided by GA returned a classification matrix that shows the correct classifications by type, as well as, the incorrect classifications by type (see Table 5). The standard ANN did not produce this level of specificity.

The primary reason for the ANN guided by GA to perform better is the objective function which is maximization of the correct classification rate as opposed to minimizing average squared error which is the objective function of the standard ANN. In the context of network detection in which classifying the connection is the most important criteria, maximizing the correct

classification rate is a more applicable objective function. However, in most standard BP network, the objective function deals with minimizing the squared error. Therefore, the ANN guided by GA is a better data mining method in network detection purposes.

As noted earlier, the ANN guided by GA provided a detailed result (see Table 5) which can be beneficial to network administrators. However, the question then becomes 'Is such detail necessary?' The ability to know where, and how, the classification method failed is important if the method is to be improved. In the case of network intrusion, a simpler method may detect that a connection attempt is an attack, but it may misclassify the type of attack (ex. probe attack classified as a DOS attack). On the positive side, the probe attack has still been detected. On the negative side, the method may miss the next probe attack because its classification pattern is incorrect. Without the detail provided by the ANN guided by GA classification matrix as shown in Table 5, this misclassification would go unnoticed.

From decision making perspective, using the methods discussed here, IT managers will be able to detect various attacks and decide about network security mechanisms and implementation strategies. As mentioned earlier, the cost of network intrusion is alarmingly high. The ability to make quick decisions regarding network attacks will help the IT managers to reduce the cost of intrusion significantly. In that regard, data mining methods discussed here could be vital for IT managers.

LIMITATIONS AND FUTURE RESEARCH

Although this study makes a contribution to the literature on network intrusion classification, there are also a number of limitations. The most significant limitation is the inability to incorporate other data mining methods, such as decision trees, memory-based systems, logistic regression, and discriminant analysis. One of the reasons for not including the other methods is that most of these methods (e.g., logistic regression) can only provide two-group classification. But one of our objectives in this study was to classify network attacks in to multiple groups. However, the lack of a full comparison across different data mining methods limits the ability to draw firm conclusions and leaves some doubt as to the validity of the conclusions that were drawn.

Future research in this area is definitely needed. Future studies could create customized algorithms which employ the classification methods used here, but produce results that are more comparable. Also, other data mining methods should be incorporated. Studies could also be conducted with more attack types which are totally new, or variations on existing types. In this vein, other studies could address the problem of classifying rarely seen attack types like the U2R and R2L. Because of their rarity these attacks are hard to develop patterns for and consequently difficult to detect, as was seen in our classification results.

CONCLUSION

Network intrusion is not a problem which will be going away. On the contrary, research indicates that it will be getting significantly worse (Zhu et al., 2001). Based on the need for networks to detect attacks, and the difficulty associated with detection, this paper has proposed the expanded use of data mining to address the problem. The ability of data mining to extract patterns from existing data makes it an excellent choice for use in intrusion detection. Both methods tested in this paper performed well in classifying the training data. However, the ANN guided by GA performed much better with the validation data. Considering that the validation data had 14 additional attack types not present in the training data, these results show the potential value of using the ANN guided by GA method in classifying network intrusions.

REFERENCES

1. Berry, M. J. A., and Linoff, G. (1997) Data Mining Techniques for Marketing, Sales, and Customer Support. New York, John Wiley & Sons, Inc.
2. Bonafacio, J. M., Cansian, A. M., Carvalho, A. C. P. L. F., and Moreira, E. S. (1997) Neural Networks Applied in Intrusion Detection Systems, *International Conference on Computational Intelligence and Multimedia Applications*, Gold Coast, Australia. 276-280
3. Caudill, M. (1991) Neural network training tips and techniques, *AI Expert*, 6, 1, 56-61.
4. Cavusoglu, H., Mishra, B., and Raghunathan, S. (2002) Assessing the Value of Detective Control in IT Security, *Eigth Americas Conference on Information Systems*, Dallas, TX. 1910-1918
5. Cavusoglu, H., Raghunathan, S., and Mishra, B. (2002) Optimal Design of Information Technology Security Architecture, *Twenty-Third International Conference on Information Systems*, Barcelona, Spain. 749-756

6. Chin, S. K. (1999) High Confidence Design for Security, *Communication of the ACM*, 42, 7, 33-37.
7. Chung, H. M., and Gray, P. (1999) Special Section Editorial: Data Mining, *Journal of Management Information Systems*, 16, 1, 11-16.
8. Debar, H., Becker, M., and Siboni, D. (1992) A Neural Network Component for An Intrusion Detection System. *1992 IEEE Symposium on Research in Security and Privacy*, Oakland, CA, IEEE Computer Society Press. 240-250
9. Fayyad, U. M., Piatetsky-Shapiro, G., and Smyth, P. (1996) From Data Mining to Knowledge Discovery in Databases, *AI Magazine*, 17, 3, 37-54.
10. Fox, K. L., Henning, R. R., Reed, J. H., and Simonian, R. P. (1990) A Neural Network Approach towards Intrusion Detection, *13th National Computer Security Conference*, Washington, D.C., NIST. 125-134
11. Lee, W., Stolfo, S. J., and Mok, K. W. (2000) Mining Audit Data to Build Intrusion Detection Models, *Fourth International Conference on Knowledge Discovery and Data Mining*, Menlo Park, CA, AAAI Press. 66-72
12. Lippmann, R., and Cunningham, R. K. (2000) Improving Intrusion Detection Performance Using Keyword Selection and Neural Networks, *Computer Networks*, 34, 597-603.
13. Lukasik, H. (2000) Protecting the Global Information Commons, *Telecommunication Policy*, 24, 6-7, 519-531.
14. Lunt, T. F. (1993) A Survey of Intrusion Detection Techniques, *Computer and Security*, 12.
15. Mehta, M., and George, B. (2001) Security in Today's E-World, *Seventh Americas Conference on Information Systems*, Boston, MA. 1219-1226
16. Messier, W. F., and Hansen, J. V. (1988) Inducing Rules for Expert System Development: An Example Using Default and Bankruptcy Rules, *Management Science*, 34, 12, 1403-1415.
17. Newmann, P. (1999) Information Systems Adversities and Risks, http://www.oas.org/juridico/english/information_systems_adversities_a.htm.
18. Power, R. (2002) CSI/FBI Computer Crime and Security Survey, *Computer Security Issues and Trends*, 8, 1, 1-22.
19. Sung, T. K., Chang, N., and Lee, G. (1999). "Dynamics of Modeling in Data Mining: Interpretive Approach to Bankruptcy Prediction." *Journal of Management Information Systems*, 16, 1, 63-85.
20. Tam, K., and Kiang, M. (1992) Managerial Applications of Neural Networks: The Case of Bank Failure Prediction, *Management Science*, 38, 7, 926-947.
21. Ullman, R., and Ferrera, D. (1998) Crime on the Internet, *Boston Bar Journal*, Nov/Dec, 6.
22. Weiss, S. M., and Kapouleas, I. (1989) An Empirical Comparison of Pattern Recognition, Neural Nets, and Machine Learning Classification Methods, *Eleventh International Joint Conference on Artificial Intelligence*, San Francisco. 781-787.
23. Zhu, D., Premkumar, G., Zhang, X., and Chu, C. (2001) Data Mining for Network Intrusion Detection: A Comparison of Alternative Methods, *Decision Sciences*, 32, 4, 635-660.