

A Statistical Approach to TCP Session Classification

Tudor Moscalu, Andrew M. Steel, Edward D. L. Brown, Yangkind L. Lim

Abstract— Government computer networks need a real-time network traffic monitoring tool to detect anomalies in network traffic patterns to improve security. Specifically, they need a tool to determine if a host is using a network connection for something other than the intended use. A key step in developing this tool is creating statistical models to accurately identify the application protocols of sessions in a network without relying on port numbers, which conventionally identify them.

This paper outlines the construction of these models. Specifically, it focuses on the methods used to build them, which included: structuring network data in a database, aggregating packet level data into sessions, and then identifying the key variables. The models employ variables such as the inter-arrival time between packets, the variance of those times, the distribution of TCP control flags and other information available from the packet headers. The paper examines the significance of these explanatory variables and attempts to determine which would be useful in a real-time implementation.

I. INTRODUCTION

EVERY day information is transferred over the Internet in the form of packets. These packets are becoming increasingly difficult to classify in terms of the purpose they serve. Malicious users have become more sophisticated over the years and are constantly inventing new methods to disguise their actions. Because of this, there is an increasing need to correctly classify network sessions in order to provide adequate network security.

This project answers the call for additional research in the field of computer network security. The work for this report was conducted within the scope of a fourth-year Systems Engineering capstone project that focused on computer network traffic classification. Current network intrusion detection systems identify and stop threats based on matching traffic patterns to known attack definitions and communication port mappings [1]. This intrusion detection method cannot adapt to new attacks and clever port masquerading [2]. The team implemented methods using statistics to classify packet flows. The significance of a project like this is an improvement in the security of network services because administrators will be able to better distinguish malicious traffic from normal traffic.

Manuscript received April 7, 2008.

T. Moscalu is with the University of Virginia.

A.M. Steel is with the University of Virginia
(e-mail: asteel@virginia.edu)

E. D. L. Brown is with the University of Virginia.

Y. L. Lim is with the University of Virginia.

II. PROJECT OVERVIEW

A. Objective

Given the current challenges of changing network security environments, the objective of the capstone project team was to develop classification tree models which could be used to correctly identify communication session server ports by looking at session variable characteristics. The development of such models validates the work of previous researchers and opens the door for the creation of new firewall and intrusion detection technology.

B. Client

The client for this project is David Marchette, principal scientist at the Naval Surface Warfare Center. The capstone team was comprised of four undergraduate systems engineering students. Statistical research with network traffic data had been previously conducted by James P. Early, Carla E. Brodley, and Catherine Rosenberg [3]. In their work the researchers were able to successfully identify 82% to 100% of the network sessions without making use of communication port numbers. The results were encouraging; however, the scope of their work was limited by the use of an artificially manufactured data set.

C. Assumptions

A major concept in machine learning is the idea that an algorithm is either supervised or unsupervised. An unsupervised learning algorithm could theoretically classify Internet traffic correctly without the assumption that the data from which it generates the model is free from abnormalities and that it is representative of the application. Because the data that the team used came from reputable sources inside firewalls maintained by IT experts, the team will assume that the port number is representative of the application with which it is normally associated or at least that the number of anomalies is so small that they would not affect the model. For example, any TCP session taking place on a server over port 80 will be assumed to be web traffic.

III. DATA AGGREGATION

The capstone team used three data sets in the analysis process. These data sets were comprised of data collected by George Mason University during 2003 and 2004 (GMU dataset), enterprise network data collected and anonymized by Lawrence Berkeley National Laboratory as part of its LBNL/ICSI Enterprise Tracing Project (Enterprise dataset), and data collected by the capstone team on a small local

computer network comprised of ten to twelve users (House dataset). The Enterprise data set consisted of eleven gigabytes of packet header information stored in multiple anonymized libpcap files. To process this binary data, a custom C++ tool was created using a free C++ libpcap library to read the captured files and import them into a MySQL database. Non-TCP packets were removed in the import process to eliminate additional network noise. Batch files were written to allow for an unsupervised upload process. A similar process was employed with the GMU and House datasets. The packet import process yielded three new MySQL tables containing 129,903,861 packets (Enterprise), 7,024,590 packets (GMU), and 1,110,335 packets (House) respectively. Each packet had twenty variables associated with it, corresponding to the information found in the packet headers.

To complete the data aggregation process, sessions were extracted using a custom, group designed session aggregation tool. In order to correctly rebuild the sessions, this tool followed TCP specific rules concerning sequence numbers and sliding windows. For each session it computed averages, variances, packet totals, and tracked protocol flag usage. The reader should note that packets containing duplicate application were deleted from the analysis. This was done as an attempt to separate network behavior from application behavior as much as possible. A foreign key was added to the packet table to associate each packet with its appropriate session in order to allow for the use of time series analysis techniques, an approach that is promising given our results so far. The session aggregation process yielded three new tables containing 453,135 sessions (Enterprise), 91,016 sessions (GMU), and 21,311 sessions (House) respectively. Each session had forty-four variables associated with it.

TABLE I
TOP ENTERPRISE APPLICATIONS

Port	Frequency	Application
80	247802	HTTP
443	32951	HTTPS
515	30696	Printer Daemon
25	25459	SMTP
139	13634	Netbios
445	12847	Microsoft – ds
993	10789	IMAPS
135	10261	Epmmap
389	7874	LDAP
3396	6172	Printer Agent
631	4242	Internet Printing Protocol
143	3241	IMAP
1521	3128	nCube License Manager
110	3114	POP3
22	2990	SSH
1026	2616	Calendar Access Protocol

Looking at the TCP sessions that were collected from the packet data, one can easily see that the network behavior of the three data sets varies with the network on which it was collected. For example, in the Enterprise data set the main application that was used was HTTP, whereas at GMU the

two main applications were HTTP and Microsoft SQL Server. The House data is also very different in that many of the applications are media related and HTTPS is used more frequently than HTTP. This may be because many of the websites used for school purposes are via secure HTTP. Another interesting thing to note is the use of gnutella on the research network of GMU. The fact that the networks have very different behavior would suggest that there might not be a general model good enough for all internet traffic but only specific to certain subnets.

TABLE II
TOP GMU APPLICATIONS

Port	Frequency	Application
80	47788	HTTP
1433	32316	Microsoft SQL Server
443	3531	HTTPS
25	2205	SMTP
3531	499	Joltid
6346	464	gnutella-svc
3127	415	CTX Bridge Port
110	295	POP3
5131	221	Unknown
5190	212	AOL
25959	207	Unknown
1850	206	GSI
8200	186	TRIVNET
3967	145	PPS Message Service
554	2616	Real Time Streaming Protocol
4662	124	OrbitNet Message Service

TABLE III
TOP HOUSE APPLICATIONS

Port	Frequency	Application
443	13326	HTTPS
80	7582	HTTP
5190	47	AOL
110	16	POP3
143	15	IMAP
6346	12	Gnutella-svc
4000	8	Terabase
38189	6	Unknown
5050	4	Multimedia Conference Control Tool
1935	4	Macromedia Flash Communications Server MX
45777	3	Unknown
26089	3	Unknown
19880	2	Unknown
16691	2	Unknown
6640	2	Unknown
25	2	SMTP

After logically grouping the packets into their respective sessions the team did some preliminary analysis into field types that would be good indicators of application type. Instead of looking at what variables would be good predictors the team first analyzed which ones would not be good. Originally, the team hypothesized that the length of the IP header could yield information about different types of options that certain applications employed. However, all of the IP headers had a length of 20 bytes, which is the default length without any options, making it completely

useless as a predictor.

The team then looked at the proportion of TCP flags, which are used to control the transmission of application data. Some of the flags rarely, if ever, get used so they were not a good indicator of application type. Also, some flags are used in almost every packet, so they were not a good predictor either. The only flag that seemed to be varied enough in its use was the push flag, which tells the receiver to push the data up to the application layer as soon as possible. Whether or not this variable was a good predictor was left up to the model building algorithm.

Another value on which the team collected statistics was the time to live (TTL) field in the IP header. The team hypothesized that certain applications would choose different values for this field; however, upon analysis it was discovered that there are really only two main choices: 128 and 64. This led the team to conclude that this would not be an effective predictor either; however, in some of the models the TTL was significant. This may be because there were not as many ports in the House data as in the Enterprise data. The rest of the statistics concerning inter-arrival times, duration and amounts of data were deemed to good potential predictors, especially after looking at factor plots which separated the data into the application ports.

The team also noticed some interesting interaction effects in the data. Sometimes a client would establish a TCP connection with a server and then the server would send 1 byte of data to the client without having requested anything. The majority of these cases took place over port 515, which is a printing process.

IV. CLASSIFICATION TREE MODELING AND TESTING

Classification tree models represent a simple way to predict categorical responses based on a given set of predictor variables. The variable that best distinguishes application protocols would form the top node of the decision tree. In Fig. 3, let us assume that feature A is the mean inter-arrival time and Class 1 is HTTP traffic. Then, if a flow has a mean inter-arrival time that is significantly greater than a pre-determined threshold it is classified as HTTP traffic. Otherwise, another decision is made that further distinguishes the flow from other types of application protocols.

Classification tree models were used to predict the server port value based on thirty-eight communication session variables. This was accomplished through the use of the Insightful Miner software package. Specifically, the 'Classification Tree Module' was used. The module makes use of the recursive partitioning code RPART developed by Terry M. Therneau, Professor of Biostatistics at the Mayo Clinic, and Beth Atkinson, Assistant Professor of Biostatistics at the Mayo Clinic [4]. RPART uses a greedy method to scan the data and select the best possible splits [4].

To build the models, the network session data was

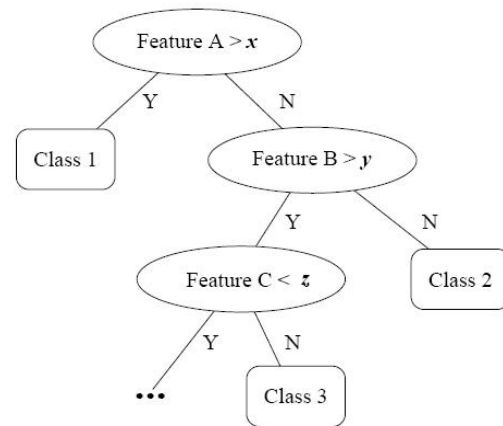


Fig. 1[3]

exported from the database and imported into IMiner, a program with the ability to handle large data sets such as ours. Separate analyses were performed for the complete sessions and all sessions (complete and incomplete). In order to assess the model's ability to predict correctly outside of the data which was used to fit the model, we randomly split the data into training and testing data sets. The training data consisted of seventy percent of the total observations while the testing data contained the remaining thirty percent. Finally, we obtained twelve working files which resulted from the following combinations: 3 (Enterprise / GMU / House) x 2 (complete and complete + incomplete) x 2 (testing and training).

The training data sets were used to build the classification trees while the testing data sets were used to determine accuracy. Confusion matrices were used to track the accuracy of each model on a port by port basis. Each constructed model was tested against all of the test sets available, regardless of dataset or session status (complete or incomplete). Thus, a model built for the Enterprise sessions was tested against test sets from both the GMU and House data sets. This was done to examine the robustness of the model across independent data sets.

The team further refined the model building and testing process by creating new training and testing data containing a select number of ports based on their occurrence frequency within the data set and relevance to daily network usage. This removed minority classes and noise from the data. As a final step new testing and training data sets were constructed by selecting sessions with ports for web traffic (80), encrypted web traffic (443), and e-mail traffic (25 & 110). The main goal for creating these data sets was to test the robustness of the classification tree models across independent data sets among commonly used ports. Successful results within these comparisons would suggest that classification tree models could be used to classify general network traffic successfully regardless of the environment on which they were trained.

V. CLASSIFICATION TREE RESULTS

The group created a total of 18 decision trees based on the training datasets. There are six models for each of the three original datasets; half of these models are created from data consisting of completed sessions; the other half is based on all sessions. The models are further segregated by the ports that they predict on. Models predict all the ports existing in the original datasets, the top ports, or the four common ports (e.g. http, https, smtp, and pop). The results represented in this paper will be separated based on the ports which they predict.

A. Data Contains All Ports

The first set of classification tree models was built using all the server port values. As it can be seen in Table IV classification tree accuracy varied between 91.9% and 98.8% for models including all the sessions, complete and incomplete, and between 92.7% and 99.3% for models including only complete sessions. Given the results it appears that the classification tree algorithms performed well when tested within the same data set. However, performance varied significantly when the models are tested against other data sets. It can be observed that the Enterprise data models do poorly when benchmarked against the GMU data sets and better when benchmarked against the House data. It is the opinion of the project team that this is largely due to the limited number of ports used in the House and GMU data sets.

TABLE IV
ALL PORTS

Training Data Set	A	B	C	D	E	F
EnterpriseAll	91.9	92.7	86.9	74.3	97.8	98.2
EnterpriseComplete	90.2	93.8	57.8	73.8	98.8	99
GMUAll	69	70.1	95.5	94.1	98.2	98.5
GMUComplete	71.8	69.7	62.3	93.7	45.7	55.3
HouseAll	45.1	42.1	49.4	28.3	98.8	99.3
HouseComplete	56.9	55.6	54.2	31.5	99.1	99.3

A: % correct against Enterprise test set with incomplete sessions
 B: % correct against Enterprise test set with only complete sessions
 C: % correct against GMU test set with incomplete sessions
 D: % correct against GMU test set with only complete sessions
 E: % correct against House test set with incomplete sessions
 F: % correct against House test set with only complete sessions

B. Data Contains Only Most Frequent Ports

The results obtained in Table IV spawned the idea to look at a more refined subset of data comprised of top ports across the data sets. Completing this process, however, was daunting as the Enterprise data contained port values not present in the other two data sets. Given this issue, models were built and tested only within their own data set. The results obtained are presented in Table V. They show a significant improvement in accuracy for the Enterprise data set; however, they remain largely unchanged for the other two data sets. The improvement in the Enterprise data set is largely due to the removal of minority classes in terms of port values. This allows the classification tree algorithm to

improve its accuracy by having fewer classification categories to select and classify.

TABLE V
TOP PORTS

Training Data Set	A	B	C	D	E	F
EnterpriseAll	95.3	96.2	-	-	-	-
EnterpriseComplete	96.2	93.7	-	-	-	-
GMUAll	-	-	98.8	98.9	-	-
GMUComplete	-	-	96.3	93.7	-	-
HouseAll	-	-	-	-	99.5	99.7
HouseComplete	-	-	-	-	99.3	99.7

A: % correct against Enterprise test set with incomplete sessions
 B: % correct against Enterprise test set with only complete sessions
 C: % correct against GMU test set with incomplete sessions
 D: % correct against GMU test set with only complete sessions
 E: % correct against House test set with incomplete sessions
 F: % correct against House test set with only complete sessions

Further, false positives were recorded for the Enterprise all session model. From Table VI it is evident that ports 80, 1026, 23, 3396 have high levels of accuracy. The number of false positives gives insight into the level of performance one could expect if the model was implemented into an intrusion detection system. If the administrator was trying to keep people from using port 80 for anything other than web traffic, about 1 percent of the time such activity would not be detected.

TABLE VI
TOP PORTS FALSE POSITIVES

Port	Frequency	Application
80	990	1.35%
445	357	10.09%
443	1164	12.87%
631	20	1.83%
515	692	7.53%
1026	8	1.19%
389	221	10.48%
22	126	25.30%
143	118	17.15%
139	584	16.11%
110	64	8.50%
135	166	5.96%
25	901	12.74%
1521	167	20.69%
993	195	6.67%
23	0	0.00%
3396	64	3.67%
21	26	13.20%

C. Data Contains Only Four Ports

Additionally, ports 25, 80, 110, and 443 were selected to construct new classification tree models. The new models were built and tested across all the available data sets. The scoring results for these models can be seen in Table VII. The overall predictive power of the new models varied between 73.9% and 99.5%. The models also exhibited predictive power increases across the data sets when compared with the results in Table IV. The only exception occurred in the Enterprise models when tested against the House data set. The most robust model observed was the GMU all session model which scored between 94.2% and

99.4% across all the data sets. Another interesting feature observed in the results for the four port models was the high level of predictive accuracy for http traffic (Port 80). This can be observed in Table VIII.

TABLE VII
TOP FOUR PORTS

Training Data Set	A	B	C	D	E	F
EnterpriseAll	98.5	98.9	93.6	84	78.3	-
EnterpriseComplete	98.2	99	90.6	83.6	73.9	-
GMUAll	94.2	94.7	99	98.5	99.4	-
GMUComplete	88.1	89.1	96	98.1	93.8	-
HouseAll	81.4	80	91.7	79.5	99.5	-
HouseComplete	-	-	-	-	-	-

A: % correct against Enterprise test set with incomplete sessions
 B: % correct against Enterprise test set with only complete sessions
 C: % correct against GMU test set with incomplete sessions
 D: % correct against GMU test set with only complete sessions
 E: % correct against House test set with incomplete sessions
 F: % correct against House test set with only complete sessions

TABLE VIII
PORT 80

Data Set	A	B	C	D	E	F
EnterpriseAll	99.4	99.6	97.6	98.9	96.8	-
EnterpriseComplete	99.2	99.6	99.3	99.7	99.5	-
GMUAll	98.4	98.6	99.7	99.8	99.6	-
GMUComplete	91.4	91.7	98.4	99.4	89.7	-
HouseAll	89.7	90.2	97.4	97	99.1	-
HouseComplete	-	-	-	-	-	-

A: % correct against Enterprise test set with incomplete sessions
 B: % correct against Enterprise test set with only complete sessions
 C: % correct against GMU test set with incomplete sessions
 D: % correct against GMU test set with only complete sessions
 E: % correct against House test set with incomplete sessions
 F: % correct against House test set with only complete sessions

VI. CONCLUSIONS

A. Classification Trees

Regardless of the number of ports used, classification tree models achieved prediction accuracy levels of 90% and 100% when tested against matching test sets. This result implies that when given a data set containing past behavior, IMiner classification tree models can be used successfully to predict traffic type on a given network. This factor could potentially allow for the creation of network specific filters to help IT professionals identify threats. The results confirm and advance Early et al.'s prior work with the C5.0 classification tree algorithm [3]. By not limiting port number values and not sampling equal numbers of ports for each port value initially, a more realistic network production environment was simulated. Additionally, the models used all the session variables available in the prediction process.

The classification tree models were also benchmarked against multi-categorical logistical models. Those models were built using the LOGISTIC function in SAS and used the same training and testing data sets as the classification tree models. The port prediction results for the multi-

categorical logistical models for top ports can be seen in Table IX. Overall they indicate that classification tree models performed better across all data sets and indicate that the logistic models were not complex enough to handle port prediction. Additionally, these models required significant processing time which imposes limits on their usability both in an offline or online data analysis environment.

TABLE IX
MULTI-CATEGORICAL LOGISTIC PREDICTION TOP PORTS

Data Set	A	B	C	D	E	F
EnterpriseAll	63.9	-	-	-	-	-
EnterpriseComplete	-	-	-	-	-	-
GMUAll	-	-	90.9	-	-	-
GMUComplete	-	-	-	97	-	-
HouseAll	-	-	-	-	97.7	-
HouseComplete	-	-	-	-	-	96.5

A: % correct against Enterprise test set with incomplete sessions
 B: % correct against Enterprise test set with only complete sessions
 C: % correct against GMU test set with incomplete sessions
 D: % correct against GMU test set with only complete sessions
 E: % correct against House test set with incomplete sessions
 F: % correct against House test set with only complete sessions

The results from the models with four ports seem to indicate that classification tree models for specific common network ports behave in robust ways across independent data sets. A good example of this is the prediction accuracy levels for Port 80 (Table VIII). However, the accuracy of these results are questioned when the Gini Index charts are reviewed. The Gini Index is a goodness of fit measure / impurity measure which is computed over the entire tree to determine a variable's importance. The aim of each split in a tree is to lower the overall value of the Gini Index statistic [5]. Further work in this area should be conducted with more data sets which are similar in size but independent in nature.

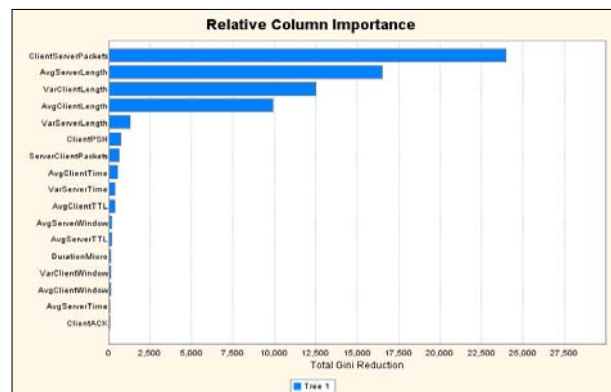


Fig. 2 Enterprise All Sessions Four Port Gini Index Variable Ranking

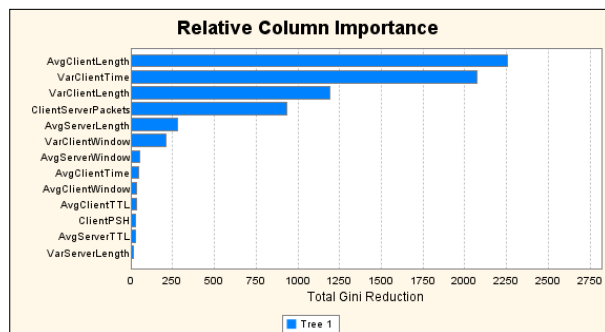


Fig. 3 GMU All Sessions Four Port Gini Index Variable Ranking

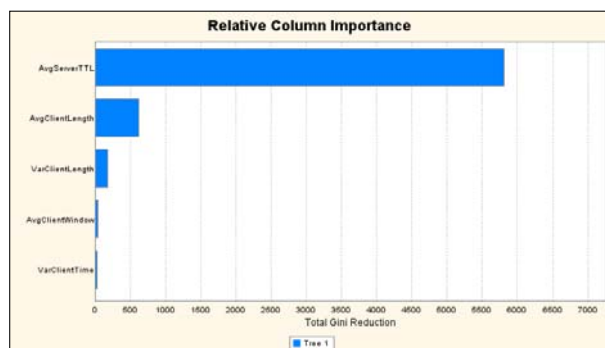


Fig. 4 House All Sessions Four Port Gini Index Variable Ranking

Selecting only top port values did improve the prediction results for the Enterprise data set. This approach could be used to reduce false positives for commonly used ports on a network. However, in terms of real time classification, this approach would not save any significant time versus an approach which includes all the port values in the model creation process. This is because once a model has been created it is extremely easy to navigate.

The results for models with complete and incomplete sessions were also similar in nature because the session aggregation software generated session variables by averaging the characteristics of all the packets for the session regardless of whether or not it was complete or incomplete. By looking at the data sets it appears that incomplete sessions are part of normal network traffic and should be included in the model generation process.

B. Network Analysis Toolkit and Other Future Work

It is important to note that a vast majority of the work conducted for this project revolved around developing successful ways to capture and manage large network binary capture files. The software tool developed by the project team to convert and load the network sessions data into a MySQL server represents a major accomplishment. The tools created can serve as a launching platform for future capstone research and Intrusion Detection Systems (IDS). It is the belief of the team that a successful IDS will need to be able to store live network traffic data in order to maintain a current training data set for model creation. This will allow any type of predictive model to self-learn and adapt over

time.

Due to time constraints, there are several aspects of the project that the team was not able to complete. These include suggestions from Prof. Carla Brodley that detailed a method to determine the optimal model for a classification tree which takes into account higher-level interactions between variables. Essentially, it is very similar to the idea of a stepwise method for determining an optimal regression model where one drops or adds a variable to get the greatest increase in maximum likelihood. She also suggested a method to deal with the vast dissimilarity of the size of the categories. For example, HTTP represents the large majority of the traffic collected from the network. Therefore, the models are going to be biased towards predicting those classes of traffic because it yields the highest accuracy. She proposed two methods of doing so: replicating the data belonging to the minority class to force the decision tree algorithm to place more weight on those observations or randomly selecting a sample from the larger data set to achieve the same result.

C. Final Thoughts

The report author and capstone team firmly believe that the results obtained are an important step in the right direction for the creation of a self learning session traffic prediction Intrusion Detection System. The model building procedures laid forward here were an exercise in model prediction accuracy and were not aimed directly at identifying anomalous traffic. Further research should be performed with more realistic, evenly balanced and independent data sets.

Finally, a live implementation of an IMiner classification tree algorithm should be attempted by future Network Analysis Capstone project teams. This could be accomplished by making use of the XML model export feature available in the IMiner software.

ACKNOWLEDGMENT

The team would like to thank Professor Ginger Davis for her support as the technical advisor throughout the year.

REFERENCES

- [1] J. Erman, A. Mahanti, M. Cohen, and C. Williamson, "Offline/realtime traffic classification using semi-supervised," in *Performance Evaluation*, vol. 64, no. 9-12, pp. 1194-1213, 2007.
- [2] A. Patcha, and J. Park, "Network anomaly detection with incomplete audit data," in *Computer Networks: The International Journal of Computer and Telecommunications Networking*, vol. 51, no. 13, pp. 3935-3955, 2007.
- [3] J. Early, C. Brodley, and C. Rosenberg. "Behavioral Authentication of Server Flows," In *Proc. Annual Computer Security Applications Conference*, 2003.
- [4] Insightful Cooperation, *Insightful Miner 8 User Guide: Classification Trees*. Seattle WA, 2006. Available at: <http://www.insightful.com/support/iminer80/uguide.pdf>
- [5] R. Derrig, and L. Francis, "Distinguishing the forest from the TREES: A Comparison of Tree Based Data Mining Methods," in *Casualty Actuarial Forum: Data Management Call Papers*, 2006.