Prediction and Detection of Malware Using Association Rules

[1] **Mr.B.Dwarakanath**, [2]**Mr.A.Suthakar**.
[1] M.Tech CSE – (Assistant Professor), Hindustan University
E-mail: dwarakanath@hindustanuniv.ac.in,
[2] M.Tech (IT) - Student, Hindustan University
E-mail: suthakar.asr@gmail.com

Abstract

Now it has been big challenge to detect the malware files from the computer systems. In previous work an automatic and robust tool to analyze and classify the file samples is needed so had developed an intelligent malware detection system (IMDS) by adopting associative classification method based on the analysis of application programming interface (API) execution calls. Despite its good performance in malware detection, IMDS still faces the following two challenges: 1) handling large set of the generated rules to build the classifier and 2) finding effective rules to build for classifying new file samples. In this paper, we first systematically evaluate to effects of the post processing techniques (e.g., rule pruning, rule ranking, and rule selection) of associative classification in malware detection, and then, propose an effective way, i.e., CIDCPF, to detect the malware from the "gray list". CIDCPF adapts the post processing techniques as follows: first applying Chi-square testing and insignificant rule pruning followed by using database coverage based on the Chi-square testing and insignificant rule pruning followed by using Database coverage based on the Chi-square measure rule ranking mechanism and Pessimistic error estimation, and finally performing prediction by selecting the best First rule. We have incorporated the CIDCPF method into our existing IMDS system, and we call the new system as CIMDS system. In particular, our CIMDS system can greatly reduce the number of generated rules, which makes it easy for our virus analysts to identify the useful ones. In this paper we are going to extend the CIMDS system from the following aspects, collect more detailed information about the API calls, such as their dependencies and timestamps and use it for better malware detection. We will investigate such as frequent structure mining to capture the complex relationships among the API calls, predict the types of malware. Our CIMDS currently only provides binary predictions, i.e., whether a PE files is malicious or not. A natural extension is to predict the different types of malware.

Keywords— Associative classification, malware detection, postprocessing, rule pruning, rule ranking, rule selection.

1.INTRODUCTION

Malware is software designed to damage a computer system without the owner's knowledge (e.g., viruses, backdoors, spyware, Trojans, and worms). Gather information that leads to loss of privacy or exploitation, gain unauthorized access to system resources, and other abusive behavior. Numerous attacks made by the malware pose a major security threat to computer users. Computer viruses are programs that must be triggered or somehow executed before they can infect your computer system and spread to others. Examples include opening a document infected with a "macro virus," booting with a diskette infected with a "boot sector" virus, or double-clicking on an infected program file. Viruses can then be spread by sharing infected files on a diskette, network drive, or other media, by exchanging infected files over the Internet via e-mail attachments, or by downloading questionable files from the Internet.

Malware detection is one of the computer security topics that are of great interest. Currently, the most important line of defense against malware is antivirus programs, such as Norton, MacAfee, and Kingsoft's Antivirus. These widely used malware detection software tools use signature-based method to recognize threats. Signature is a short string of bytes, which is unique for each known malware so that future examples of it can be correctly classified with a small error rate. However, this classic signature-based method always fails to detect variants of known malware or previously unknown malware, because the malware writers always adopt techniques like obfuscation to bypass these signatures. In order to remain effective, it is of paramount importance for the antivirus companies to be able to quickly analyze variants of known malware and previously unknown malware samples. Unfortunately, the number of file samples that need to be analyzed on a daily basis is constantly increasing [19]. According to the virus analysts at Kingsoft Antivirus Laboratory, the "gray list" that is needed to be analyzed per day usually contains more than 70 000 file samples. Clearly, there is a need for an automatic, efficient, and robust tool to classify the "gray list".Recently, many post processing techniques, including rule pruning, rule ranking, and rule selection have been developed for associative classification to reduce the size of the classifier and make the classification process more effective and accurate. Systematically evaluate the effects of the post processing techniques in malware detection and propose an effective way, i.e., CIDCPF, to detect the malware from the "gray list." CIDCPF adapts the post processing techniques as follows: first applying chi-square testing and insignificant rule pruning followed by using database coverage based on the chi-square measure rule ranking mechanism and pessimistic error estimation, and finally predicting the new file sample by the way of selecting the best first rule. We have incorporated the CIDCPF method into our existing IMDS system, and we call the new system as CIMDS system.

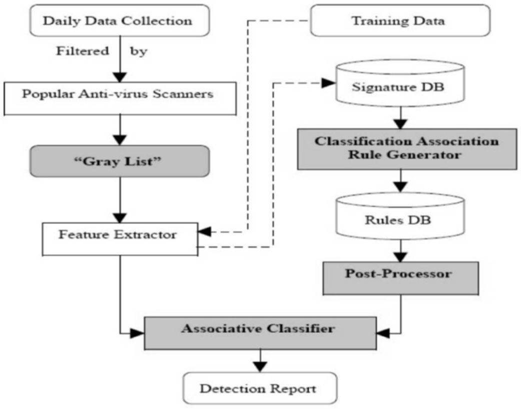

Fig 1 System Architecture

We systematically evaluate the effects of the postprocessing techniques in malware detection and propose an effective way, i.e., CIDCPF, to detect the malware from the "gray list." CIDCPF adapts the postprocessing techniques as follows: first applying chi-square testing and insignificant rule pruning followed by using database coverage based on the chi-square measure rule ranking mechanism and pessimistic error estimation, and finally predicting the new file sample by the way of selecting the best first rule. We have incorporated the CIDCPF method into our existing IMDS system, and we call the new system as CIMDS system. Case studies are performed on the large collection of file samples obtained from theAntivirus Laboratory atKingsoft Corporation and promising experimental results demonstrate that the efficiency and ability of detecting malware from the "gray list" of our CIMDS system outperform popular antivirus software, such as McAfee VirusScan and Norton AntiVirus, as well as previous data-miningbased detection systems, which employed Naive Bayes, support vector machine (SVM), and decision tree techniques. In particular, our CIMDS system can greatly reduce the number of generated rules, which makes it easy for our virus analysts to identify the useful ones.

II.RELATED WORK

In order to overcome the disadvantages of the widely used signature-based malware detection method, data mining and machine-learning approaches are proposed for malware detection. Naive Bayes method, SVM, and decision tree classifiers are used to detect new malicious executables in previous studies. Associative classification, as a new classification approach integrating association rule mining and classification, becomes one of the significant tools for knowledge discovery and data mining. Due to the fact that frequent itemsets (sets of API calls) discovered by association mining can well represent the underlying semantics (profiles) of malware and benign file datasets, associative classification has been successfully used in the IMDS system developed for malware detection. However, there is often a huge number of rules generated in a classification association rule mining practice. It is often infeasible to build a classifier using all of the generated rules. Hence, how to reduce the number of the rules and select the effective ones for prediction is very important for improving the classifier's ACY and efficiency. Recently, many postprocessing techniques, including rule pruning, rule ranking, and rule selection have been developed for associative classification to reduce the size of the classifier and make the classification process more effective and accurate. It is interesting to know how these postprocessing techniques would help the associative classifiers for malware detection. In this paper, we systematically evaluate the effects of the postprocessing techniques in malware detection and propose an effective way, i.e., CIDCPF, to detect the malware from the "gray list."

Programs that have the potential to invade privacy and security of system are given a term Potentially Unwanted Programs (PUP). These programs include virus, Spy ware, adware, Trojan, worms. These programs may compromise confidentiality, integrity, and availability of the system or may obtain sensitive information without the user's consent. There are many commercial inducements also which serve as fertile land to the industry to flourish and there will be an increase in PUP in future. In start, virus was the only malicious threat and since then much research has been done in this area. A more recent type of malicious threat is Spyware. According to the University of Washington's department of computer science and Engineering, Spy ware is defined as "software that gathers information about use of a computer, usually without the knowledge of the owner of the computer, and relays the information across the Internet to a third party location". Another definition of Spy ware is given as "Any software that monitors user behavior, or gathers information about the user without adequate notice, consent, or control from the user". Spyware may be capable of capturing keystrokes, taking screenshots, saving authentication credentials, storing personal email addresses and web form data, and thus may obtain behavioral and personal information about users. This can lead to financial loss, as in identity theft and credit card fraud .The knowledge about Spy ware is generally perceived as low among the common users and the process of Spyware identification or removal is generally considered as outside of their competence. Spyware may show characteristics like nonstop appearance of advertisement pop-ups. It may open a website or force the user to open a website which has not been visited before, install browser toolbars without seeking acceptance from the user, change search engine results, make unexpected changes in the browser, and display error messages. Furthermore, indications of Spyware include a noticeable change in computer speed after installation of new software, auto opening of software or browser, a changed behavior of already installed software, network traffic without request, and increased disk utilization even in idle situations. Some researchers have doubtingly predicted that advanced Spyware can possibly take control of complete systems in the near future. There is no single anti-Spyware tool that can prevent all existing Spyware because without vigilant examination of a software package, the process of Spyware detection has become almost impossible. Spyware can be a part of freeware, plug-in, shareware, or illegal software. Normally, one would need a diverse set of anti-Spyware software to be fully protected. Anti-virus program may not be capable of detecting the Spyware until it has been designed for this purpose. Current anti-virus systems use signature-based methods or heuristic-based approaches against different malware. Signature-based Anti-virus systems use specific features or unique strings extracted from binary code. This method demonstrates good results for known viruses but lacks the capability of identifying new and unseen malicious code. Heuristic-based systems try to detect known and unknown Malware on the basis of rules defined by experts who define behavior patterns for malicious and benign software. The heuristic method is considered costly and often ineffective against new Spyware. A heuristic approach, on the other hand, may detect novel threats with a reasonable accuracy. Anti-virus software is normally not designed with the focus on spyware but some experiments are done to prove that they can be used for Spyware detection. Consequently, we cannot be sure that they are capable of detecting new types of Spyware. So it may be possible to apply some other existing technologies that can help in finding new Spyware. A new approach that can be used for the detection of Spyware is data mining. Data mining is widely adopted in various fields such as weather forecasting, marketing campaigns, discovering patterns from the financial data for fraud detection, etc. Data mining uses historical data for the prediction of a possible outcome in future. Data mining is an application of machine learning that is a subarea of Artificial Intelligence (AI). Machine learning is a study of making a system intelligent that learns automatically to make correct predications or to act intelligently without human assistance. Machine learning encompasses with different fields especially statistics but mathematics and computer science as well. It has applied data mining approach for the detection of worms and built a classification model which secured 94.0 % of overall accuracy with random forest classifier. Many Spyware are considered legal but yet could be dangerous to the computer systems. In 2005, the US Federal Trade Commission (FTC) prosecuted Seismic Entertainment Productions and stopped them infecting consumer PCs with Spyware. According to the commission they had developed a method that detained control of computers nationwide by spreading Spyware and other malicious software and by flooding advertisements to their clients, this breach had made computers work slowly or stopped them from working. In the end Seismic released their anti-Spyware software to counter all problems that they themselves had created and earned more money than what had been earned previously by spreading the Spyware.

III. CLASSIFICATION ASSOCIATION RULE GENERATION

Associative classification, as a new classification approach integrating association rule mining and classification, becomes one of the significant tools for knowledge discovery and data mining. It can be effectively used in malware detection, since frequent itemsets are typically of statistical significance and classifiers based on frequent pattern analysis are generally effective to test datasets. In this section, we briefly discuss the generation of rules for classification.

A. Data Collection and Transformation

We obtain 50 000 Windows PE files of which 15 000 are recognized as benign executables and the remaining 35 000 are malicious executables. PE is designed as a common file format for all flavors of Windows operating system, and PE malicious executables are in the majority of the malware rising in recent years. All the file samples are provided by the Antivirus Laboratory of Kingsoft Corporation, and the malicious executables mainly consist of backdoors, spyware, trojans, and worms. Based on the system architecture of our previous malware detection system IMDS [35], we extract the API calls as the features of the file samples and store them in the signature database. There are six fields in the signature database, which are record ID, PE file name, file type (“0” represents benign file, while "1" is for malicious file), called APIs name, called API ID and the total number of called API functions. The transaction data can also be easily converted to relational data if necessary. Now the data is ready for classification association rule generation

B. Classification Association Rule Generation

For malware detection in this paper, the first goal is to find out how a set of API calls supports the specific class objectives:

class1 = malicious, and class2 = benign.

1) Support and confidence: Given a dataset DB, let $I = \{I1, \ldots, Im\}$ be an itemset and $I \rightarrow$ class(os, oc) be an association rulewhose consequent is a class objective.

2) Frequent itemset: Given *mos* as a user-specified minimum support. I is a frequent itemset/pattern in DB if $os \geq mos$.

3) Classification association rule: Given *moc* as a userspecified confidence. Let $I = \{I1, \ldots, Im\}$ be a frequent itemset. $I \rightarrow$ class(os, oc) is a classification association rule if $oc \geq moc$.

IV.POSTPROCESSING TECHNIQUES OF ASSOCIATIVE CLASSIFICATION FOR MALWARE DETECTION SYSTEM

The goal of our malware detection system is to build classifier using the generated rules to classify the new file samples more effectively and accurately, so the postprocessing of associative classification is very important for improving the system's ACY and efficiency. The postprocessing techniques includes rule pruning, rule ranking, and rule selection.

A. Rule Pruning Approaches

Accompanied with the ability of mining the complete set of the rules, associative classification also has a major drawback that the number of generated rules can be really large and the removal of the redundant or misleading rules is indispensable. Besides the five common rule pruning approaches introduced in Section II: 1) $\chi 2$ (chi-square) testing tomeasure the significance of the rule itself; 2) redundant rule pruning to discard the specific rules with fewer confidence values; 3) database coverage to just keep the rules covering at least one training data object not considered by a higher ranked rule; 4) pessimistic error estimation to test the estimated error of a new rule; and 5) lazy pruning to discard the rules incorrectly classifying the training objects, we here propose another rule pruning method before building the classifier, named "insignificant rules pruning." Since many generated rules are redundant or minor variations of others and their existence may simply be due to chance rather than true correlation, these insignificant rules should be removed.

B. Rule Ranking Mechanisms

Within the associative classification framework, regardless of which particular methodology is used to generate the rules, a classifier is usually represented as an order list of the generated rules based on some rule ranking mechanisms. Many associative classification algorithm. utilize rule ranking procedures as the basis for selecting the classifier during pruning and later for predicting new data objects. As we discussed in Section II, there are five common ranking mechanisms: CSA, ACS, WRA, Laplace accuracy, and $\chi 2$ measure. Here, we give a more detailed introduction.

1) CSA: Based on the well-established "support-confidence" framework, CSA first sorts the original rule list based on their confidence values in a descending order. For those rules that share a common confidence value, CSA sorts them in a descending
order based on the support values. CSA sorts the rules sharing common values for both confidence and support in an ascending order based on the size of the rule antecedent.

2) ACS: Ensuring that "specific rules have a higher precedence than more general rules", ACS considers the size of the rule antecedent as the most significant factor (using a descending order) followed by the rule confidence and support values, respectively.

3) WRA: WRA assigns an additive weighting score to each rule to determine its expected ACY. The calculation of the value
of a rule r is: WRA(r) = supp (r.antecedent)*(conf (r)-supp (r.consequent)) [4]. In the rule reordering stage, the original rule
list is sorted based on the assigned WRA value in a descending order.

4) Laplace accuracy: The principle of Laplace accuracy is similar to WRA. The calculation of the Laplace value of a rule
r is
Laplace (r) = (supp (r.antecedent $\cup$ r.consequent) + 1)
(supp (r.antecedent) + c)
where c represents the number of predefined classes.

5) $\chi 2$ measure: In associative classification algorithms, if the $\chi 2$ measure between two variables (the antecedent and consequent-class of the generated rule) is higher than a certain threshold value, we can conclude that there might be a relation between the rule antecedent and consequent-class, otherwise, it implies that the two variables may be statistically independent. We can order the list of the generated rules in a descending order based on their $\chi 2$ values. For the aforementioned five rule ranking mechanisms, we empirically study all of them for building the classifier and later for detecting the new malware.

C. Rule Selection Methods

After building the classifier by the techniques of rule pruning and rule ranking, we can select the subset of the rules from the classifier to predict the new file samples. As stated in Section II, there are three common rule selection approaches: best first rule, all rules, and best *k* rules. For our malware detection system, we will also try all of these methods to predict the new file samples and find the best way for malware detection.

V,PERFORMANCE EVALUATION

A. Experiment Setup

We randomly select 35 000 executables from our data collection, including 14 000 benign executables, 5255 backdoors, 5245 spyware, 5200 trojans, and 5300 worms in the training dataset. The rest 15 000 executables are used for testing purpose of which 6000 are benign files and 9000 are malicious ones. After filtering some of the worthless API calls, we finally extract 5102 API calls from the training dataset. By using the OOA_Fast_FP-Growth algorithm, we generate 31 rules with the minimum support and confidence as 0.18 and 0.5, respectively for the benign class, while 8424 rules are derived with the minimum support and confidence as 0.25 and 0.7, respectively for the malicious class.1 To systematically evaluate the effects of postprocessing techniques for malware detection, we conduct the following three sets of experimental studies using our collected data obtained from the Antivirus Laboratory of Kingsoft Corporation. The first set of study is to compare the ACY and efficiency of the three different associative classifier building algorithms: CBA, CMAR, and CPAR [37], when used for malware detection system. Since none of the three algorithms adopt the insignificant rule pruning approach, in the second set of study, we prune the insignificant rules before building the classifier. From these two sets of studies, we will choose the best rule pruning and rule selection methods for malware detection. In third set of experiments, we will compare the five rule ranking mechanisms and find the best ranking method for malware detection. Please note in all the experiments, rule mining, selection, and ranking are performed only within training data. From the three set of experiments, we will propose an effective classifier building method and incorporate it to our improved malware detection system CIMDS. All the experimental studies are conducted under the environment of Windows XP operating system plus Intel P4 1.83 GHz CPU and 1 GB of RAM.

B. Comparisons of CBA, CMAR, and CPAR for Malware Detection

Since the algorithms of CBA, CMAR, and CPAR have been successfully used in associative classification and represent different

kinds of postprocessing techniques for building the classifiers, in the first set of experiments, we use them for malware detection and compare their ACY and efficiency.

The datasets described in Section VI-A are used for training and testing. In this paper, we useDRandACYdefined as follows to evaluate each classifier building method.

1) True positive (TP): The number of executables correctly classified as malicious code.

2) True negative (TN): The number of executables correctly classified as benign executables.

3) False positive (FP): The number of executables mistakenly classified as malicious executables.

4) False negtive (FN): The number of executables mistakenly classified as benign executables.

5) DR: $TP/(TP + FN)$.

6) ACY: $TP + TN/(TP + TN + FP + FN)$.

The experimental results shown in Table IVindicate that CBA classifier building method performs better than the other two for

malware detection.

C. Insignificant Rule Pruning

From Table III, we observe that none of the three algorithms adopts the insignificant rule pruning approach. In this set of experiments, we prune the insignificant rules before building the classifier. From Tables IV and V, the comparisons illustrate that no matter, which algorithm we adopt, the number of rules selected for building the classifier decrease sharply by using insignificant rule pruning approach, while the DR and ACY of the classifiers remain the same or even slightly increase. In addition, the results in Table V illustrate that CBA classifier building method still achieves better performance than the other two for malware detection. From these two set of experiments, we can conclude that: 1) for rule pruning, using all of the approaches shown in Table II can achieve better performance; and 2) for rule selection, compared with the other two, best first rule selection approach performs best on our dataset. Since the classification rules are unevenly distributed, multiple rules-based prediction may not suitable for our dataset. Thus, in the following experiments, we use $\chi 2$ test, insignificant rule pruning, database coverage, and pessimistic error estimation approaches to prune the generated rules before building the classifier. For prediction, we will adopt the best first rule selection approach to detect the malware.

VI.CONCLUSION

In this paper, we systematically evaluate the effects of the postprocessing techniques (e.g., rule pruning, rule ranking, and rule selection) of associative classification in malware detection and propose an effective way, i.e., CIDCPF, to detect the malware from the “gray list.” To the best of our knowledge, this is the first paper on using postprocessing techniques of associative classification in malware detection. Experiments on a large real data collection fromAntivirus Laboratory at Kingsoft Corporation demonstrate among the most common and popular associative classification building methods, our CIDCPF method achieves better performance on detection ability and efficiency because of its concise, but effective classifier. In addition, our CIMDS system, which adopts CIDCPF method for building classifiers can greatly reduce the number of generated rules and make it easy for our virus analysts to identify the useful ones. Promising experimental results demonstrate that the efficiency and ability of detecting the malware form the "gray list" of our CIMDS system outperform popular antivirus software, such as McAfee VirusScan and Norton AntiVirus, as well as previous data-mining-based detection systems, which employed Naïve Bayes, LIBLINEAR SVM, and Decision Tree techniques.

VII.FUTURE ENHANCEMENT

In our futurework,we plan to extend our CIMDS system from the following aspects: 1) collect more detailed information about the API calls, such as their dependencies and timestamps and use it for bettermalware detection.We will investigatemethods such as frequent structuremining to capture the complex relationships among the API calls. 2) Predict the types of malware. Our CIMDS currently only provides binary predictions, i.e., whether a PE file is malicious or not. A natural extension is to predict the different types of malware.

REFERENCES

[1] M. Antonie and O. Zaiane, "An associative classifier based on positive and negative rules," in *Proc. 9th ACM SIGMOD Workshop Res. Issues Data Mining Knowl. Discovery*, 2004, pp. 64–69.

[2] M. Antonie, O. Zaiane, and A. Coman, "Associative classifiers for medical images," in *Proc. Mining Multimedia Complex Data (LNCS)*, 2003,pp. 68–83.

[3] B. Arunasalam and S. Chawla, "CCCS: A top-down associative classifier for imbalanced class distribution," in *Proc. KDD-2006*, 2010, pp. 517– 522.

[4] E.Baralis, S.Chiusano, and P.Graza, "On support thresholds in associative classification," in *Proc. ACM Symp. Appl. Comput. 2004*, pp. 553–558.

[5] E. Baralis and P. Torino, "A lazy approach to pruning classification rules," in *Proc. IEEE Int. Conf. Data Mining 2002*, pp. 35–42.

[6] R.Bayardo, R. Agrawal, andD.Gunopulos, "Constraint-based rule mining in large, dense databases," in *Proc. ICDE-1999*, pp. 188–197.

[7] H. Cheng, X. Yan, J. Han, and P. S. Yu, "Direct discriminative pattern mining for effective classification," in *Proc. ICDE-2008*, pp. 169–178.

[8] H. Cheng, X. Yan, J. Han, and C. Hsu, "Discriminative frequent pattern analysis for effective classification," in *Proc. ICDE-2007*, pp. 716–725.

[9] M. Christodorescu, S. Jha, and C Kruegel, "Mining specifications of malicious behavior," in *Proc. ESEC/FSE-2007*, pp. 5–14.

[10] F. Coenen and P. Leng, "An evaluation of approaches to classification rule selection," in *Proc. 4th IEEE Int. Conf. Data Mining 2004*, pp. 359–362.

[11] H. Toivonen, M. Klemetinen, P. Ronkainen, K. Hatonen, and H. Mannila, "Pruning and grouping discovered association rules," in *Proc. MlnetWorkshop Statist.,Mach. Learning, and DiscoveryDatabases*, 1995, pp. 47–52.

[12] X. Jiang and X. Zhu, "vEye: Behavioral footprinting for self-propagating worm detection and profiling," *Knowl. Inf. Syst.*, vol. 18, no. 2, pp. 231–262, 2009.

[13] J. Kolter and M. Maloof, "Learning to detect malicious executables in the wild," in *Proc. KDD-2004*, pp. 470–478.

[14] Ng. R. T. Lakshmanan and J. L. Han, "Exploratory mining and pruning optimizations of constrained association rules," in *Proc. SIGMOD-1998*, pp. 13–24.

[15] W. Li, J. Han, and J. Pei, "CMAR: Accurate and efficient classification based on multiple class-association rules," in *Proc. IEEE Int. Conf. Data Mining 2001*, pp. 369–376.

[16] B. Liu, W. Hsu, and Y. Ma, "Integrating classification and association rule mining," in *Proc. 4th Int. Conf. Knowl. Discovery Data Mining1998*, pp. 80–86.

[17] B. Liu, W. Hsu, and Y. Ma, "Pruning and summarizing the discovered associations," in *Proc. KDD-1999*, pp. 125–134.

[18] M. Mahta, R. Agrawal, and J. Rissanen, "SLIQ: A fast scalable classifier for data mining," in *Proc. EDBT-1996*, pp. 18–32.

[19] M. Bailey, J. Oberheide, J. Andersen, Z. M. Mao, F. Jahanian, and J. Nazario, "Automated classification and analysis of internet malware," in *Proc. RAID 2007, LNCS*, vol. 4637, pp. 178–197.

[20] J. R. Quinlan, *C4.5. Programs for Machine Learning.* San Mateo, CA: Morgan Kaufmann, 1993.

[21] M. Schultz, E. Eskin, and E. Zadok, "Data mining methods for detection of new malicious executables," in *Proc. Security Privacy, 2001. Proc. 2001 IEEE Symp.*, May 14–16, pp. 38–49.

[22] W. Snedecor and W. Cochran, *Statistical Methods*, 8th ed. Iowa City, IA: Iowa State Univ. Press, 1989.