Combining File Content and File Relations for Cloud Based Malware Detection

Yanfang Ye
Comodo Security Solutions
Beijing, 100082, P.R.China
yanfang@comodo.com

Tao Li
School of Computer Science
Florida International University
Miami, FL, 33199, USA
taoli@cs.fiu.edu

Shenghuo Zhu
NEC Laboratories America
Cupertino, CA, 95129, USA
zsh@sv.nec-labs.com

Weiwei Zhuang
Xiamen University
Xiamen, 361005, P.R.China
zhuangweiwei@gmail.com

Egemen Tas, Umesh Gupta
Comodo Security Solutions
New Jersey, NJ, 07310, USA
{egemen,umesh}@comodo.com

Melih Abdulhayoglu
Comodo Security Solutions
New Jersey, NJ, 07310, USA
melih@comodo.com

ABSTRACT

Due to their damages to Internet security, malware (such as virus, worms, trojans, spyware, backdoors, and rootkits) detection has caught the attention not only of anti-malware industry but also of researchers for decades. Resting on the analysis of file contents extracted from the file samples, like Application Programming Interface (API) calls, instruction sequences, and binary strings, data mining methods such as Naive Bayes and Support Vector Machines have been used for malware detection. However, besides file contents, relations among file samples, such as a "Downloader" is always associated with many Trojans, can provide invaluable information about the properties of file samples. In this paper, we study how file relations can be used to improve malware detection results and develop a file verdict system (named "Valkyrie") building on a semi-parametric classifier model to combine file content and file relations together for malware detection. To the best of our knowledge, this is the first work of using both file content and file relations for malware detection. A comprehensive experimental study on a large collection of PE files obtained from the clients of anti-malware products of Comodo Security Solutions Incorporation is performed to compare various malware detection approaches. Promising experimental results demonstrate that the accuracy and efficiency of our Valkyrie system outperform other popular anti-malware software tools such as Kaspersky AntiVirus and McAfee VirusScan, as well as other alternative data mining based detection systems. Our system has already been incorporated into the scanning tool of Comodo's Anti-Malware software.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning; D.4.6 [**Operating System**]: Security and Protection - Invasive software

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
KDD'11, August 21–24, 2011, San Diego, California, USA.
Copyright 2011 ACM 978-1-4503-0813-7/11/08 ...$10.00.

General Terms

Algorithms, Experimentation, Security

Keywords

cloud based malware detection, file content, file relation, semi-parametric model for learning from graph

1. INTRODUCTION

1.1 Cloud Based Malware Detection

Malware is software designed to infiltrate or damage a computer system without the owner's informed consent (e.g., virus, worms, trojans, spyware, backdoors, and rootkits) [23]. Numerous attacks made by the malware pose a major security threat to Internet users [8]. Hence, malware detection is one of the internet security topics that are of great interest [4, 25, 13, 15, 19, 20, 27, 24]. Currently, the most significant line of defense against malware is anti-malware software products, such as Kaspersky, MacAfee and Comodo's Anti-Malware software. Typically, these widely used malware detection software tools use the signature-based method to recognize threats. Signature is a short string of bytes, which is unique for each known malware so that its future examples can be correctly classified with a small error rate.

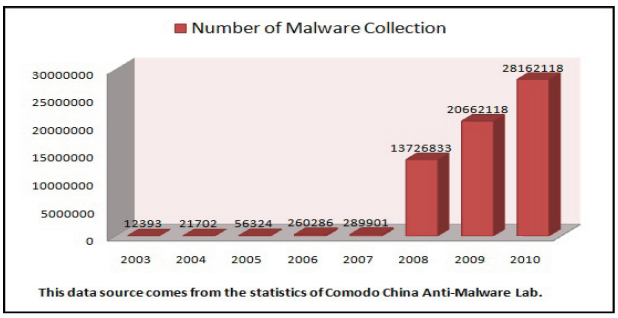

Figure 1: The Increment of Malware Samples (Data source: Comodo China Anti-Malware Lab).

However, driven by the economic benefits, malware writers quickly invent counter-measures against proposed malware analysis tech-

IPR2023-00124
CrowdStrike EX1015 Page 1

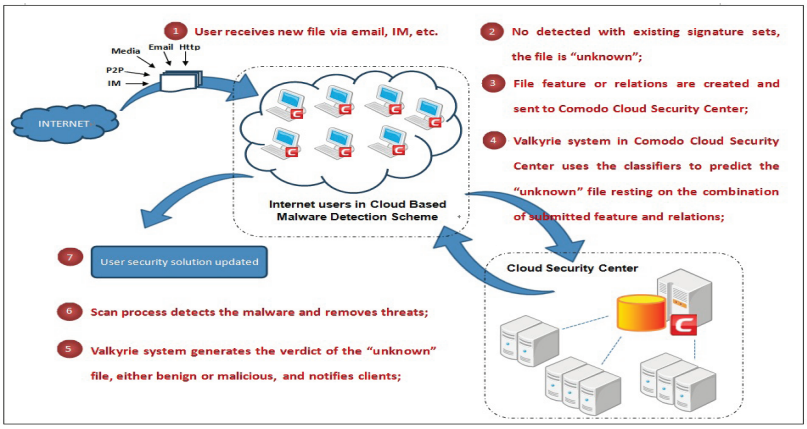

Figure 2: The Workflow of Comodo Cloud Based Malware Detection Scheme.

niques, chief among them being automated obfuscation [20]. Because of automated obfuscation, today's malware samples are created at a rate of thousands per day. Figure 1 shows the increasing trend of malware samples in P.R.China from Year 2003 to Year 2010 (this data is provided by Comodo China Anti-Malware Lab). It can be observed that the number of malware samples has increased sharply since 2008. In fact the number of malware samples in 2008 alone is much larger than the total sum of previous five years.

Nowdays malware samples increasingly employ techniques such as polymorphism [2], metamorphism [1], packing, instruction virtualization, and emulation to bypass signatures and defeat attempts to analyze their inner mechanisms [20]. In order to remain effective, many Anti-Malware venders have turned their classic signature-based method to ***cloud*** (server) based detection. The work flow of the cloud based detection method adopted by Comodo Security Solutions Incorporation is shown in Figure 2.

The work flow of this cloud based malware detection scheme can be described as follows:

1. On the client side, users may receive new files from emails, media or IM(Instant Message) tools when they are using the Internet.

2. Anti-malware products will first use the signature set on the clients for scanning. If these new files are not detected by existing signatures, then they will be marked as "unknown".

3. In order to detect malware from the unknown file collection, file features (like file content as well as the file relations) are extracted and sent to Comodo Cloud Security Center.

4. Based on these collected features, the classifier(s) on the cloud server will predict and generate the verdicts for the unknown file samples, either benign or malicious.

5. Then the cloud server will send the verdict results to the clients and notify the clients immediately.

6. According to the response from the cloud server, the scanning process can detect new malware samples and remove the threats.

7. Due to the fast response from the cloud server, the client users can have most up-to-date security solutions.

To sum-up, using the cloud-based architecture, malware detection is now conducted in a client-server manner: authenticating valid software programs from a whitelist and blocking invalid software programs from a blacklist using the signature-based method at the client (user) side, and predicting any unknown software (i.e., the gray list) at the cloud (server) side and quickly generating the verdict results to the clients within seconds. The gray list, containing unknown software programs which could be either benign or malicious, was usually authenticated or rejected manually by malware analysts before. With the development of the malware writing techniques, the number of file samples in the gray list that need to be analyzed by malware analysts on a daily basis is constantly increasing. For example, the gray list collected by the Anti-Malware Lab of Comodo Security Solutions Incorporation usually contains about 500,000 file samples per day. Therefore, there is an urgent need for anti-malware industry to develop intelligent methods for efficient and effective malware detection at the cloud (server) side.

Recently, many research efforts have been conducted on developing intelligent malware detection systems [4, 25, 13, 15, 19, 20, 27, 24]. In these systems, the detection process is generally divided into two steps: *feature extraction* and *classification*. In the first step, various features such as Application Programming Interface (API) calls [27] and program strings [13, 19, 18] are extracted to capture the characteristics of the file samples. In the second step, intelligent classification techniques such as decision trees [17], Naive Bayes, and associative classifiers [24, 13, 19, 27] are used to automatically classify the file samples into different classes based on computational analysis of the feature representations. These intelligent malware detection systems are varied in their use of feature representations and classification methods. For example, IMDS [27] performs association classification on Windows API calls extracted from executable files, while Naive Bayes methods on the extracted strings and byte sequences are applied in [19].

These intelligent techniques have isolated successes in classifying particular sets of malware samples, but they have limitations that leave a large room for improvement. In particular, none of these techniques have taken the relationships among file samples into consideration for malware detection. Simply treating file programs as independent samples allows many off-the-shelf classification tools to be directly adapted for malware classification. However, the relationships among file samples may imply the interdependence among them and thus the usual ***i.i.d*** (independent and identical distributed) assumption may not hold for malware sam-

IPR2023-00124
CrowdStrike EX1015 Page 2

ples. As a result, ignoring the relations among file samples is a significant limitation of current malware classification methods.

1.2 Relations Among File Samples

For malware detection, the relations among file samples provide invaluable information about their properties. Here we use some examples for illustration. Based on the collected file lists from clients, we construct a co-occurrence graph to describe the relations among file samples. Generally, two files are related if they are shared by many clients (or equivalently, file lists). As shown in Figure 3, we can observe that the file "yy(1).exe" is associated with many trojans which are marked as purple color. Actually, this "yy(1).exe" file is a kind of Trojan-Downloader malware. Trojan-Downloader refers to any malicious software that downloads and installs multiple unwanted applications of adware and malware from remote servers. Malware samples of this type are spread from malicious websites or by emails as attachments or links, and are installed secretly without the user's consent. Therefore, from the relations shown in Figure 3, we can infer that if an ***unknown*** file always co-occurs with many kinds of trojans in users' computers, then most likely, it is a malicious Trojan-Downloader file.

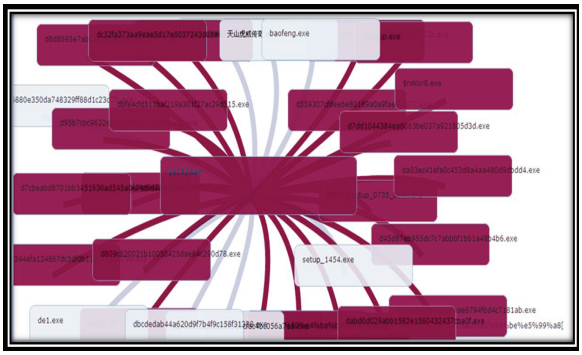

Figure 3: File Relations Between a Trojan-Downloader and its Related Trojans.

Another example showing the relations among benign files is illustrated in Figure 4. From Figure 4, we can observe that an unknown file "everest.exe" can be possibly recognized as benign since it is always associated with known benign files marked in green color. Actually, this "everest.exe" is a benign system diagnostic application which always co-occurs with its related Dynamic Link Library files, such as, "everest_start.dll", "everest_mondiag.dll", "everest_rcs.dll" and so on.

Sometime it is not easy to determine whether a file is malicious or not solely based on file content information itself. According to the experience and knowledge of our anti-malware experts, file relations among samples can be a novel and practical feature representation for malware detection. Some malware samples may have stronger connections with benign files than malicious ones. In such cases, those file samples might be infected files. Actually, these unexpected relations can be filtered and removed, because the infected samples can be detected independently using the infected file detector which is developed by our anti-malware experts.

1.3 Combining File Content and File Relation

To improve the performance of file sample classification for malware detection, in this paper, we utilize both file content and file relation information. However, relation information and file content have different properties. Relation information provides a graph structure in the data and induces pairwise similarity between objects while the file content provides inherent characteristic information about the file samples. Although both the relation information and file content can be used independently to classify file samples, classification algorithms that make use of them simultaneously should be able to achieve a better performance.

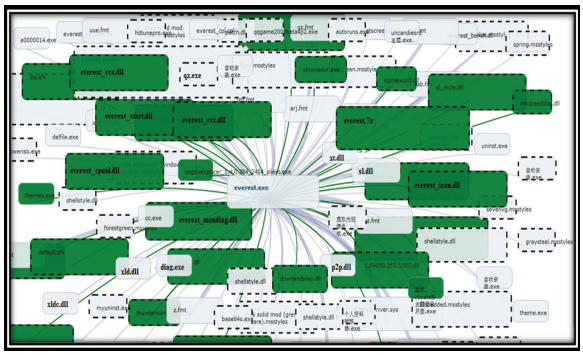

Figure 4: File Relations Between a Benign Application and its Related Dynamic Link Library files.

The problem of combining content information and relation information (i.e., link information) have been widely studied for web document categorization in data mining and information retrieval community [26, 9]. The approaches for combining content and link information generally fall into two categories: (1) feature integration which treats the relation information as additional features and enlarges the feature representation [3, 11, 16]; and (2) Kernel Integration which integrates the data at the similarity computation or the Kernel level [10, 14]. However, both types of approaches have limitations: feature integration may degrade the quality of information as file relations and file content typically have different properties, while kernel integration fails to explore the correlation and the inherent consistency between the content information and the relation information [31].

1.4 Contributions of Our Paper

In this paper, we propose a semi-parametric classification model for combining file content and file relations. The semi-parametric model consists of two components: a parametric component reflecting file content information and a non-parametric component reflecting file relation information. The model seamlessly integrates these two components and formulates the classification problem using the graph regularization framework. Our model can be viewed as an extension of recently developed joint-embedding approaches which aims to seek a common low-dimensional embedding via joint factorization of both the content and relation information [31, 5, 30]. However, different from the joint-embedding approaches, our model does not explicitly infer the embedding and is directly optimized for classification. We develop a file verdict system (named "Valkyrie") using the proposed model to integrate file content and file relations for malware detection. To the best of our knowledge, this is the first work of using both file content and file relations for malware detection. In short, our developed Valkyrie system has the following major traits:

- *Novel Usage of File Relation:* Different from previous studies for malware detection, we not only make use of file content, but also use the file relations for malware detection.
- *A Principled Model for Combining File Content and File Relations:* We propose a semi-parametric classification model

IPR2023-00124
CrowdStrike EX1015 Page 3

to seamlessly combine file content and file relation, and formulate the classification problem using the graph regularization framework.

- *A Practical Developed System for Real Industry Application:* Based on 37,930 clients, we obtain 30,950 malware samples, 225,830 benign files and 434,870 unknown files from Comodo Cloud Security Center. We build a practical system for malware detection and provide a comprehensive experimental study.

All these traits make our Valkyrie system a practical solution for automatic malware detection. The case studies on large and real daily malware collection from Comodo Cloud Security Center demonstrate the effectiveness and efficiency of our Valkyrie system. As a result, our Valkyrie system has already been incorporated into the scanning tool of Comodo's Anti-Malware software.

1.5 Organization of The Paper

The rest of this paper is organized as follows. Section 2 presents the overview of our Valkyrie system. Section 3 describes the feature extraction and representation; Section 4 introduces the proposed semi-parametric model combining file content and file relations together for malware detection; In Section 5, using the daily data collection obtained from Comodo Cloud Security Center, we systematically evaluate the effectiveness and efficiency of our Valkyrie system in comparison with other proposed classification methods, as well as some of the popular Anti-Malware software such as Kaspersky and NOD32. Section 6 presents the details of system development and operation. Section 7 discusses the related work. Finally, Section 8 concludes the paper.

2. SYSTEM ARCHITECTURE

Figure 5 shows the system architecture of our Valkyrie system. We briefly describe each component below.

- **Training:**

  1. **User File List and File Sample Collector:** It collects the file lists from the clients which contain the potential relations between file samples, together with the file samples.
  2. **File Content Feature Exactor:** Besides its high extraction efficiency compared with dynamic feature representation methods, Application Programming Interfaced (API) calls can well reflect the behaviors of program code pieces. Therefore, our developed file content feature extractor extracts the API calls from the collected malicious and benign Windows Portable Executable (PE) files. (See Section 3.1 for details.)
  3. **File Relation Feature Exactor:** Based on the collected file lists from clients, a co-occurrence graph is constructed to describe the file relations. Note that many unexpected relations (like relations between infected samples and benign files) are removed using infected file detectors. (See Section 3.2 for details.)
  4. **Semi-Parametric Model Based Classifier:** Our proposed semi-parametric model integrates file content and relation information and formulates the classification problem using the graph regularization framework. (See Section 4 for details.)

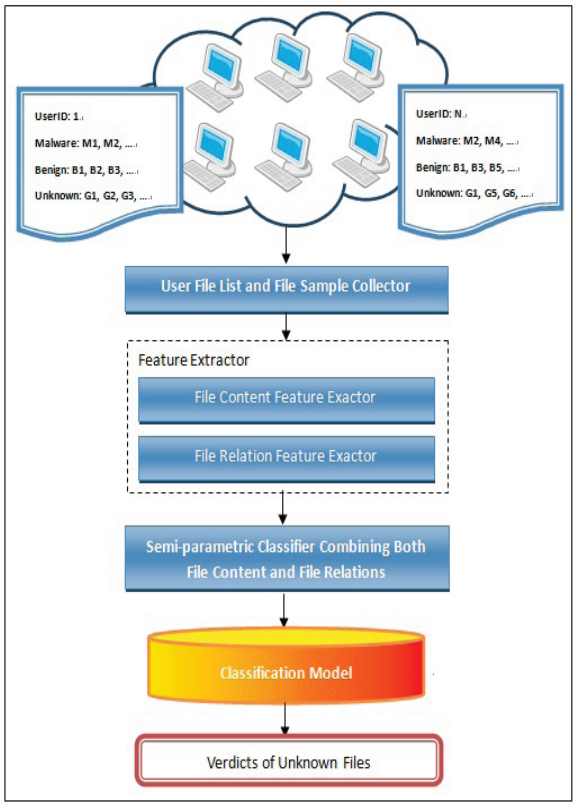

Figure 5: The System Architecture of Valkyrie.

- **Prediction:** On the clients, our Comodo Anti-Malware software products authenticate valid software from a whitelist and block invalid software from a blacklist using the signature-based method. The gray list, containing unknown software programs which could be either normal or malicious, is then fed into our Valkyrie system. After file content and file relation feature extractions, the semi-parametric model is applied to the gray list for prediction.

3. FEATURE EXTRACTION

Our Valkyrie system is performed directly on Windows Portable Executable (PE) codes. PE is designed as a common file format for all flavor of Windows operating system, and PE malware are in the majority of the malware rising in recent years [27]. In this section, we will introduce both file content and file relation feature extraction methods we adopted.

3.1 File Content

We extract the Application Programming Interface (API) calls from the Import Tables [27] of collected malicious and benign PE files, convert them to a group of 32-bit global IDs (for example, the API "MAPI32.MAPIReadMail" in encoded as 0X00000F12) as the content features, and stores these features in the signature database. A sample file content signature database is shown in Figure 6, in which there are 6 fields: record ID, PE file name, file type ("-1" represents benign file while "1" is for malicious file), called APIs name, called API ID, the total number of called API functions.

3.2 File Relations

Based on the collected file lists from clients, we construct a

IPR2023-00124
CrowdStrike EX1015 Page 4

Resultset 1

id	filename	filesort	apiseq	intvectorofapi	apifunno
9198	Adware. Casino. N__27e	1	wsock32. dll, wsock32	5, 7, 10, 12, 13,	158
9199	Adware. DropSpam__29b	1	ws2_32. dll, wsasock	1, 7, 9, 14, 15, 1	145
9201	Adware. NaviPromo. Gen	1	kernel32. dll, getcoi	6, 15, 18, 43, 45	124
9202	Backdoor. AFCore. Drop	1	kernel32. dll, exittl	2, 44, 45, 81, 13	29
9203	Backdoor. Agent. AAS__	1	kernel32. dll, getpr	1, 5, 7, 8, 10, 13	98
9209	Backdoor. Generic. 121	1	kernel32. dll, rtlze	12, 13, 39, 53, 5	74
9210	Backdoor. Hupigon. 115	1	kernel32. dll, findc	1, 5, 7, 8, 10, 11	205
9211	Backdoor. Hupigon. 115	1	kernel32. dll, findc	1, 5, 7, 8, 10, 11	205
9212	Backdoor. Hupigon. 115	1	kernel32. dll, findc	1, 5, 7, 8, 10, 11	205
9231	BehavesLike. Trojan. F	1	kernel32. dll, globa	1, 3, 4, 5, 6, 8, 1	232
9232	BehavesLike. Win32. Ex	1	kernel32. dll, tlsse	38, 39, 40, 41, 4	57
9233	BehavesLike. Win32. Ex	1	kernel32. dll, tlsse	38, 39, 40, 41, 4	57

Figure 6: Sample File Content Features in the Signature Database.

co-occurrence graph to describe the relations among file samples. Generally, two files are related if they are shared by many clients (or equivalently, file lists). Note that many unexpected relations (like relations between infected samples and benign files) are first removed using infected file detectors.

The co-occurrence graph is defined as $G =< V, E >$ where V is the set of file samples. Given two file samples v_i and v_j, let S_i be the set of file lists containing v_i and S_j be the set of file lists containing v_j. Then the similarity between v_i and v_j is computed as

$$sim(v_i, v_j) = \frac{|S_i \cap S_j|}{|S_i \cup S_j|}, \tag{1}$$

where $|S|$ denotes the size of a set S. If the similarity between a pair of file samples is greater than 0, then there is an edge between them and E is the set of edges between vertices.

An example graph is shown in Figure 7 illustrating the real relations between some file samples, where the size of each edge indicates its weight.

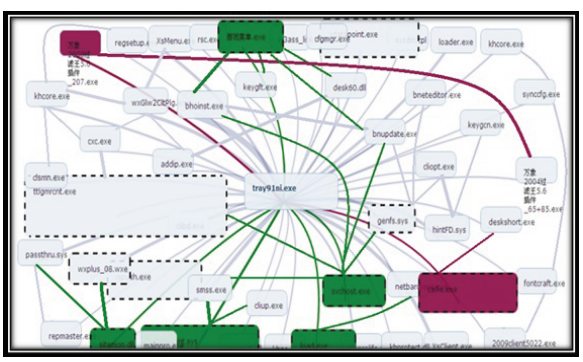

Figure 7: An example graph of real relations between some file samples (purple color-malware samples, green color-benign files, transparent color-unknown files).

4. A SEMI-PARAMETRIC MODEL FOR COMBINING FILE CONTENT AND FILE RELATIONS

In this section, we propose a semi-parametric model to combine file content and file relations for classification using the graph regularization framework. The semi-parametric model consists of two components: a parametric component reflecting file content information and a non-parametric component reflecting file relation information.

Let $\mathbf{f}$ be a vector, each of whose elements is the label (i.e., malicious or benign) of a file example to be predicted. The vector $\mathbf{f}$ can be generated from two parts, parametric and non-parametric ones. The parametric component follows a linear model, $\mathbf{X}^\top\mathbf{w}$, where each column of matrix $\mathbf{X}$ is the content feature vector of a file example, and $\mathbf{w}$ is the coefficients. The non-parametric part is just a vector of $\mathbf{h}$, each element of which corresponds to a value of a file example. Combining two parts, we have $\mathbf{f} = \mathbf{X}^\top\mathbf{w} + \mathbf{h}$.

Now considering the labeling information vector $\mathbf{y}$. Let $y_i = 1$ if the i-th file sample is malicious, $y_i = -1$ if the i-th file sample is benign, $y_i = 0$ if the i-th file sample is unlabeled. We can use hinge loss for labeled file samples as in Support Vector Machine, or use L_2 loss for labeled data as in Least Square problems. For simplicity, we follow [29] to use L_2 loss on all data points, i.e. $\|\mathbf{y}-\mathbf{f}\|^2$. We also consider the global consistency on the co-occurrence graph [29], $\mathbf{f}^\top\mathbf{L}\mathbf{f}$, where the symmetric matrix $\mathbf{L}$ is the normalized Laplacian of the graph. Thus the total loss is

$$\tfrac{1}{2}\|\mathbf{y} - \mathbf{f}\|^2 + \frac{\alpha}{2}\mathbf{f}^\top\mathbf{L}\mathbf{f}, \tag{2}$$

where α is the weight for combining two parts of information, adding $\frac{1}{2}$ is just for convenience.

To limit the model complexity, we add the regularization terms for $\mathbf{w}$ and $\mathbf{h}$, which are

$$\frac{1}{2\beta}\mathbf{w}^\top\mathbf{w} + \frac{1}{2\gamma}\mathbf{h}^\top\mathbf{h}, \tag{3}$$

where β and γ are the regularization parameters.

Putting Eq. (2) and Eq. (3) together, we have optimization problem:

$$\min_{\mathbf{f},\mathbf{w},\mathbf{h}} \quad \tfrac{1}{2}\|\mathbf{y} - \mathbf{f}\|^2 + \frac{\alpha}{2}\mathbf{f}^\top\mathbf{L}\mathbf{f} + \frac{1}{2\beta}\mathbf{w}^\top\mathbf{w} + \frac{1}{2\gamma}\mathbf{h}^\top\mathbf{h} \tag{4}$$
$$\text{subject to} \quad \mathbf{f} = \mathbf{X}^\top\mathbf{w} + \mathbf{h}.$$

To solve Eq. (4), we introduce Lagrange multiplier $\boldsymbol{\xi}$.

$$\begin{aligned}\mathcal{L}(\mathbf{f}, \mathbf{w}, \mathbf{h}; \boldsymbol{\xi}) \quad = \quad & \tfrac{1}{2}\|\mathbf{y} - \mathbf{f}\|^2 + \frac{\alpha}{2}\mathbf{f}^\top\mathbf{L}\mathbf{f} + \frac{1}{2\beta}\mathbf{w}^\top\mathbf{w} \\ & + \frac{1}{2\gamma}\mathbf{h}^\top\mathbf{h} + \boldsymbol{\xi}^\top(\mathbf{f} - \mathbf{X}^\top\mathbf{w} - \mathbf{h}).\end{aligned}$$

As $\frac{\partial\mathcal{L}}{\partial\mathbf{w}} = 0$, $\frac{\partial\mathcal{L}}{\partial\mathbf{h}} = 0$, $\frac{\partial\mathcal{L}}{\partial\boldsymbol{\xi}} = 0$, and $\frac{\partial\mathcal{L}}{\partial\mathbf{f}} = 0$, we have

$$\mathbf{w} = \beta\mathbf{X}\boldsymbol{\xi} \tag{5}$$
$$\mathbf{h} = \gamma\boldsymbol{\xi} \tag{6}$$
$$\mathbf{f} = \mathbf{X}^\top\mathbf{w} + \mathbf{h} \tag{7}$$
$$\mathbf{y} = \mathbf{f} + \alpha\mathbf{L}\mathbf{f} + \boldsymbol{\xi} \tag{8}$$

Plugging Eqs. (5,6) into Eq. (7), we have

$$\mathbf{f} = (\beta\mathbf{X}^\top\mathbf{X} + \gamma\mathbf{I})\boldsymbol{\xi},$$

or

$$\boldsymbol{\xi} = (\beta\mathbf{X}^\top\mathbf{X} + \gamma\mathbf{I})^{-1}\mathbf{f}.$$

Plugging it into Eq. (8), we have

$$\mathbf{f} = \left[\mathbf{I} + \alpha\mathbf{L} + (\beta\mathbf{X}^\top\mathbf{X} + \gamma\mathbf{I})^{-1}\right]^{-1}\mathbf{y}. \tag{9}$$

This model is an extension of [29] by consider the parametric part. Note that if there are no content features, then $f = h$ and this

model reduces to traditional semi-supervised learning. Different from [31] and [30], this model does not infer the embedding.

Computation Issues: We need to solve

$$\left[\mathbf{I} + \alpha\mathbf{L} + (\beta\mathbf{X}^\top\mathbf{X} + \gamma\mathbf{I})^{-1}\right]\mathbf{f} = \mathbf{y} \tag{10}$$

Let the size of $\mathbf{X}$ be $p \times n$, where p is the number of feature and n is the number of instances, the average nonzeros element of $\mathbf{L}$ be κ. As long as $p \ll n$, we can follow the Woodbury identity, and Eq. (10) becomes

$$\left[(1+\gamma^{-1})\mathbf{I} + \alpha\mathbf{L} - \gamma^{-1}\mathbf{X}(\gamma\beta^{-1}\mathbf{I} + \mathbf{X}\mathbf{X}^\top)^{-1}\mathbf{X}^\top\right]\mathbf{f} = \mathbf{y} \tag{11}$$

To solve Eq. (11), we can use conjugate gradient descent method. Computing $\mathbf{X}\mathbf{X}^\top$ is $O(np^2)$, the inverse of $(\gamma\beta^{-1}\mathbf{I} + \mathbf{X}\mathbf{X}^\top)$ is $O(p^3)$, and we precompute $(\gamma\beta^{-1}\mathbf{I}+\mathbf{X}\mathbf{X}^\top)^{-1}\mathbf{X}^\top$ with $O(np^2 + p^3)$. In each iteration of conjugate gradient descent, we compute $\left[(1+\gamma^{-1})\mathbf{I} + \alpha\mathbf{L} - \gamma^{-1}\mathbf{X}(\gamma\beta^{-1}\mathbf{I} + \mathbf{X}\mathbf{X}^\top)^{-1}\mathbf{X}^\top\right]\mathbf{v}$ for some $\mathbf{v}$. The computation of each iteration is $O(n(p+\kappa))$. The convergence rate depends on the condition number of the LHS matrix of Eq. (11).

5. EXPERIMENTAL RESULTS AND ANALYSIS

In this section, we conduct three sets of experimental studies using our data collection obtained from Comodo Cloud Security Center to fully evaluate the performance of our developed Valkyrie system: (1) In the first set of experiments, we evaluate the effectiveness of file content based classifier and file relation based classifier for malware detection; (2) In the second part of experiments, we evaluate our proposed semi-parametric model based classifier by comparing it with alternative methods for combining file content and file relations. (3) In the last set of experiments, we compare our Valkyrie system with some of the popular anti-malware software products such as Kaspersky Anti-Virus, MaAfee VirusScan, Bitdefender.

5.1 Experimental Setup

We measure the malware detection performance of different classifiers using the following evaluation measures:

- **True Positive (TP)**: the number of samples correctly classified as malicious files.
- **True Negative (TN)**: the number of samples correctly classified as benign files.
- **False Positive (FP)**: the number of samples mistakenly classified as malicious files.
- **False Negtive (FN)**: the number of samples mistakenly classified as benign files.
- **Accuracy (ACY)**: $\frac{TP+TN}{TP+TN+FP+FN}$
- **Recall (RC)**: $\frac{TP+TN+FP+FN}{TheNumberOfTotalFileCollection}$.

The dataset we obtained from Comodo Cloud Security Center includes 37,930 user file lists that describe file relations between 30,950 malware samples, 225,830 benign files and 434,870 unknown files (analyzed by the anti-malware experts of Comodo Security Lab, 39,138 of them are malware, while 395,732 of them are benign files). We also have the file relation information for all the file samples. Using the feature extraction methods described in Section 3, based on this data collection, 1) resting on the API calls extracted from the known file samples, we obtain 210,850 training file content vectors (since part of the file samples' Import Table are invalid, 23,610 malicious files can be effectively extracted their API calls, while 187,240 benign samples are successfully extracted) with 86,757 dimensions; 2) from the collected user file lists, after excluding the unexpected relations (like relations between infected samples and benign files), we construct a graph including 248,986 vertices (29,006 represent malicious files, while 219,980 represent benign samples) with 356,134 edges.

All the experimental studies are conducted under the environment of Windows 7 operating system plus Intel(R) Core(TM) i3 CPU and 4 GB of RAM.

5.2 Comparisons of File Content and File Relation Based Classifiers

In this set of experiments, we evaluate the effectiveness of malware detection results based on different feature representations: file content and file relations. The large collection of file sample data along with the high dimensionality and sparseness requires the classification methods for malware detection to be scalable and robust. With the advantage of handling large feature space without overfitting, Support Vector Machine (SVM) has shown state-of-art results in classification problems [22, 12, 28]. Therefore, in this section, we use SVM [7] as the base classifier. For file content based classification, SVM is applied on the features of API calls. For file relation based classification, we treat file relations as the features for each file sample, i.e., the i-th feature is the similarities with the i-th file sample. Linear SVM [7] is used in both cases and the regularization parameter of SVM is selected using cross-validation.

From Table 1 and Figure 8, we observe that the accuracy of file relation based classifier is similar to file content based classifier, while the recall of the file relation based classifier greatly outperforms the file content based classifier for unknown file verdicts.

Training	TP	FP	TN	FN	ACY	RC
F_Content	23,585	32	187,208	25	0.9997	0.8211
F_Relation	27,018	880	219,100	1,988	0.9885	0.9696
Testing	TP	FP	TN	FN	ACY	RC
F_Content	23,358	2,230	236,196	6,423	0.9677	0.6168
F_Relation	25,969	6,880	312,100	9,988	0.9525	0.8162

Table 1: Comparisons of File Content and File Relation Based Classifiers. Remark: "F_Content"-File Content based Classifier, "F_Relation"-File Relation based Classifier.

5.3 Comparisons of Different Classifiers Combining File Content and File Relation

In this section, we compare our semi-parametric model with the following methods of combining file relations and file content information: (1) SVM on feature integration: We combine the content features and the relation features and then apply SVM on the enlarged feature space. We use different weights for these two sets of features and the weights are selected using cross-validation. (2) SVM on kernel integration: We average the linear kernel on the content and the relation similarity (note that the co-occurrence graph can be viewed as a kernel) and apply SVM on the composite kernel. (3) joint-factorization: We use the supervised joint matrix factorization on both the content and relation information and then perform SVM on the resulting low dimensional embedding. For

IPR2023-00124
CrowdStrike EX1015 Page 6

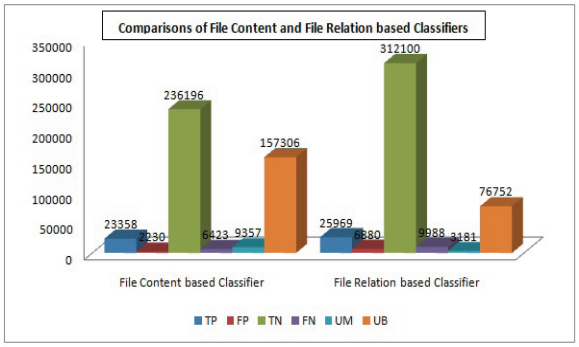

Figure 8: Comparisons of File Content and File Relation Based Classifiers. Remark: "UM"-the number of malware from unknown file collection still unrecognized by classifier, "UB"–the number of benign files from unknown file collection still unrecognized by classifier.

more details on this method, please refer to [31]. For our semi-parametric model, the parameters α, β, and γ are all set to 0.1.

The results as shown in Table 2 and Figure 9 demonstrate that: (1) Combining file relation with file content can improve the classification effectiveness for malware detection; (2) Combining file relation with file content, our proposed semi-parametric model based classifier outperforms other alternative methods.

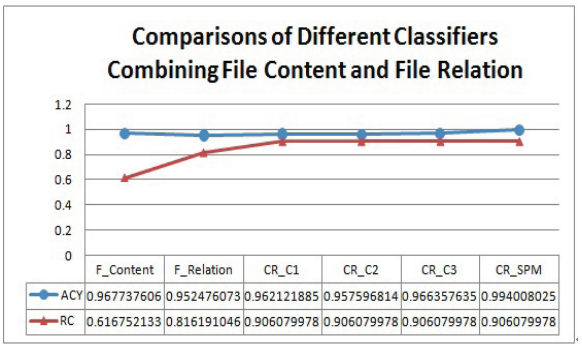

Figure 9: Comparisons of Different Classifiers Combining File Content and File Relation. Remark: "F_Content"-File Content based Classifier, "F_Relation"-File Relation based Classifier, "CR_C1"-SVM on Feature Integration, "CR_C2"-SVM on Kernel Integration, "CR_C3"-Joint-factorization Classifier, "CR_SPM"-our proposed Semi-Parametric Model.

Testing	TP	FP	TN	FN	ACY	RC
F_Content	23,358	2,230	236,196	6,423	0.9677	0.6168
F_Relation	25,969	6,880	312,100	9,988	0.9525	0.8162
CR_C1	29,002	7,454	350,100	7,471	0.9621	0.9061
CR_C2	28,123	8,358	349,196	8,350	0.9576	0.9061
CR_C3	30,789	7,572	349,982	5,486	0.9664	0.9061
CR_SPM	34,675	563	356,991	1,798	0.9940	0.9061

Table 2: Comparisons of Different Classifiers Combining File Content and File Relation. Remark: "F_Content"-File Content based Classifier, "F_Relation"-File Relation based Classifier, "CR_C1"-SVM on Feature Integration, "CR_C2"-SVM on Kernel Integration, "CR_C3"-Joint-factorization Classifier, "CR_SPM"-our proposed Semi-Parametric Model.

AV.	TP	FP	TN	FN	ACY	RC
Kasp	27,954	711	0	0	0.9752	0.0659
Nod32	26,589	923	0	0	0.9665	0.0633
Mcafee	23,951	1,011	0	0	0.9595	0.0574
BD	28,763	780	0	0	0.9736	0.0679
Avira	29,009	1,887	0	0	0.9389	0.0710
Valkyrie	34,675	563	356,991	1,798	0.9940	0.9061

Table 3: The malware detection results of different AV software products on the collection with 434,870 unknown files.

5.4 Comparisons with Different AV Venders

In this section, we apply Valkyrie system in real applications to evaluate its malware detection effectiveness and efficiency on the daily data collection described in Section 5.1.

5.4.1 Comparisons of Detection Effectiveness between Different AV Venders

Based on 434,870 unknown files (analyzed by the anti-malware experts of Comodo Security Lab, 39,138 of them are malware, while 395,732 of them are benign files), we first compare the malware detection effectiveness of Valkyrie system with some of the popular AV products, like Kaspersky(Kasp), NOD32, Mcafee, Bitdefender(BD) and Avira. For comparison purpose, we use all of the Anti-Virus scanners' latest versions of the base of signature on the same day(Feb 14th, 2011). Table 3 and Figure 10 show that the malware detection effectiveness of our Valkyrie outperforms other popular AV products based on our huge data collection.

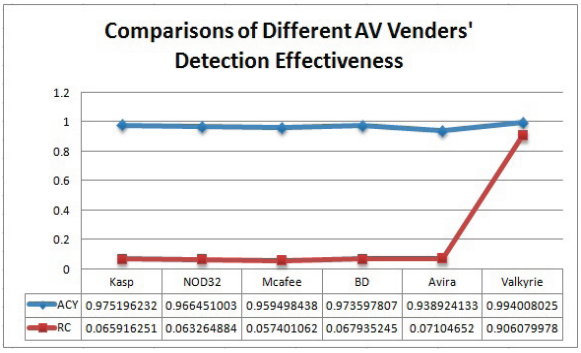

Figure 10: The comparisons of malware detection results of different AV software products on the collection with 434,870 unknown files.

5.4.2 *Comparisons of Detection Efficiency between Different AV Venders*

In this set of experiments, we compare the detection efficiency of our developed Valkyrie system with different AV scanners. The results in Figure 11 illustrate that our Valkyrie system achieves much higher efficiency than other popular scanners when being executed in the same environment, due to its high efficient feature extraction ways and novel detection scheme.

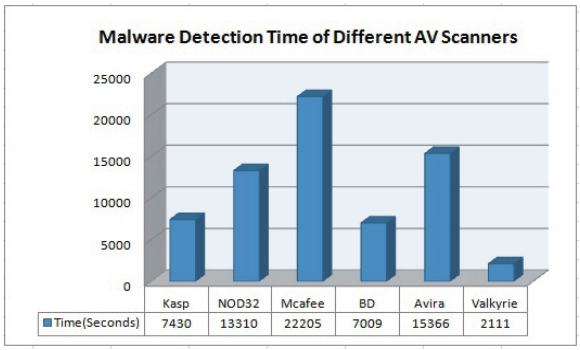

Figure 11: The comparisons of malware detection efficiency of different AV software on 434,870 file collection.

All of these traits make it possible for real anti-malware industry application. The Valkyrie system has been incorporated into the Comodo Anti-Malware products.

6. SYSTEM DEVELOPMENT AND OPERATION

Till now, by adopting our proposed classification method, the system has been extended to combine file relations and various kinds of file content representations, such as, program strings, instructions, functions extracted from export table and so on. We now totally have 15 classifiers combining file relations and 15 different file content representations. Figure 12 shows the real file verdict service (http://valkyrie.comodo.com) for pubic which integrates our developed Valkyrie system. For the development of the Valkyrie system, Comodo has spent over $550K, $150K of which is on the hardware equipment. The system monitors 15 classifiers that verify functionality and availability and is managed in a revision control system. Over 40 anti-malware analysts at Comodo Cloud Security Center are utilizing the system on the daily basis. In practice, a human analyst has to spend at least 8 hours to manually analyze 100 file samples for malware detection. Using the Valkyrie system, the analysis of about 500,000 file samples (including feature extraction and prediction) can be performed within 8 hours using 5 servers. The high efficiency of our Valkyrie system can greatly save human labors and reduce the staff cost. This would benefit over 8 million Internet users of Comodo's client anti-malware products.

7. RELATED WORK

7.1 Malware Detection

With the development of malware industry and the sharp increase of the number of malware samples, most of the Anti-Malware Venders have switched from signature-based detection methods to the cloud based scheme. In the cloud based malware detection scheme, anti-Malware Venders authenticate valid software programs from a whitelist and block invalid software programs from a blacklist using signature-based method on the clients, collect and predict a large number of unknown software programs (i.e., the gray list) on the cloud (server), and quickly generate the verdict results and send to the clients within seconds. As the number of file samples in the gray list increasing rapidly with the development of the malware writing techniques, there is an urgent need for anti-malware industry to apply intelligent techniques on the cloud side for automatic malware detection. Using these intelligent techniques, the detection process is generally divided into two steps: *content feature extraction* and *classification*. The detection performance is critically dependent on the set of extracted features and the classifier [18].

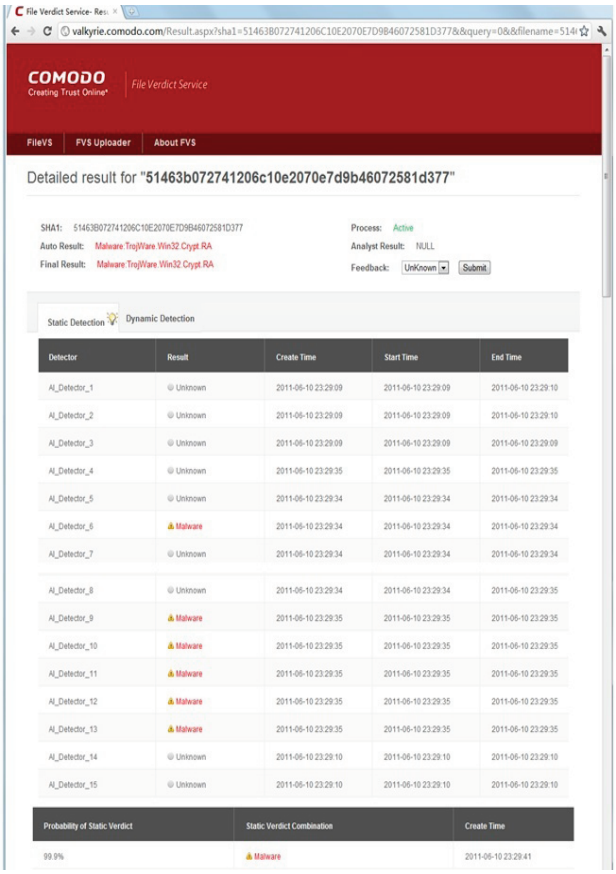

Figure 12: File Verdict Service for Pubic Integrated Valkyrie System.

Content Feature Extraction: For content feature extraction, there are mainly two types of methods [23]: static and dynamic feature extraction. Compared with dynamic feature extraction methods, static feature extraction methods are easier and less expensive [27]. In addition, according the statistics from Comodo Cloud Security Center, over 80% of the collected file samples in our work can be represented by static features, while just about 40% of the file samples can run dynamically. Therefore, we use static feature extraction in our work. There are many different kinds of static features [8], such as Application Programming Interface (API) calls, binary string, and file instructions, each has its own advantages and disadvantages. In our work, we extract Application Programming Interface (API) calls from the collected malicious and benign PE files since they can well reflect the behavior of program code pieces.

IPR2023-00124
CrowdStrike EX1015 Page 8

Classification: For classification, over the last couple of years, many data mining and machine learning approaches have been adopted for malware detection [21, 4, 25, 13, 15, 19, 20, 27, 24]. Neural Networks as well as immune system are used by IBM for computer virus recognition [21]. Naive Bayes method, Support Vector Machine(SVM), decision tree and associative classification methods are applied to detect new malicious executables in previous studies [13, 19, 24, 27].

However, all of these methods are applied on the file content features. Sometimes, according to file content itself, it is not easy to determine whether a file is malicious or not solely based on the file content. Recently, a malware detection system based on large-scale graph inference using file relationships is developed in [6]. To improve the detection performance, in our work, we use both the file content and file relation information for classification.

7.2 Combining Content Information and Link Information

The problem of combining content information and relation information (i.e., link information) have been widely studied for web document categorization in data mining and information retrieval community [26]. Early approaches for combining content information and link information fall into two general categories: (1) feature integration [3, 11, 16]: this approach enlarges the feature representation to incorporate all data and produce a unified feature space. In particular, the relation information is viewed as additional features/attributes. However, since file relations and file content have different properties, direct feature integration may degrades the quality of information. (2) Kernel Integration [10, 14]: The data is kept in their original form and they are integrated at the similarity computation or the Kernel level. In other words, relation similarity and content similarity are combined directly. One drawback of the kernel integration is that it does not fully explore the correlation and the inherent consistency between the content information and the relation information. Recently Joint-Embedding approaches are proposed to address the limitations of the above two types of approaches [31, 5, 30]. These Joint-Embedding approaches first seek a common low-dimensional embedding via joint factorization of both the content and relation information and then perform classification in the transformed space. In our work, we propose a semi-parametric classification model for combining file content and file relations. Our model can be regarded as an extension of Joint-Embedding approaches. However, different from the joint-embedding approaches, our model does not explicitly infer the embedding and is directly optimized for classification.

8. CONCLUSION

In this paper, we study how file relations can be used to improve malware detection results for a large collection of file samples and develop a file verdict system (named "Valkyrie") based on a semi-parametric classification model to combine file content and file relations together for malware detection. To the best of our knowledge, this is the first work of using both file content and file relations for malware detection. Empirical studies on large and real daily data sets collected by Comodo Cloud Security Center illustrate that our Valkyrie system outperforms other malware classification methods as well as some of the popular AV products. The system has been incorporated into the Comodo's Anti-Malware products.

Acknowledgement

The work of T. Li is partially supported by the US National Science Foundation under grants IIS-0546280 and DMS-0915110.

9. REFERENCES

[1] P. Beaucamps and E. Filiol. Metamorphism, formal grammars and undecidable code mutation. In *Journal in Computer Science, 2 (1)*, pages 70–75, 2007.

[2] P. Beaucamps and E. Filiol. On the possibility of practically obfuscating programs towards aunified perspective of code protection. In *Journal in Computer Virology, 3 (1)*, 2007.

[3] S. Chakrabarti, B. Dom, and P. Indyk. Enhanced hypertext categorization using hyperlinks. *SIGMOD Rec.*, 27:307–318, June 1998.

[4] M. Christodorescu, S. Jha, and C.Kruegel. Mining specifications of malicious behavior. In *Proceedings of ESEC/FSE'07*, pages 5–14, 2007.

[5] D. Cohn and T. Hofmann. The missing link - a probabilistic model of document content and hypertext connectivity. In *Advances in Neural Information Processing Systems 13*, 2001.

[6] D. Chau, C. Nachenberg, J. Willhelm, A. Wright and C. Faloutsos. Polonium: Tera-scale graph mining and inference for malware detection. In *Proccedings of SIAM International Conference on Data Mining (SDM) 2011*, 2011.

[7] R. Fan, K. Chang, C. Hsieh, X. Wang, and C. Lin. Liblinear: A library for large linear classification. *J. Mach. Learning Res.*, 9:1775–1778, 2008.

[8] E. Filiol. Computer viruses: from theory to applications. In *Springer, Heihelberg*, 2005.

[9] M. Fisher and R. Everson. When are links useful? experiments in text classification. In *ECIR*, 2003.

[10] T. Joachims, N. Cristianini, and J. Shawe-Taylor. Composite kernels for hypertext categorisation. In *ICML*, 2001.

[11] P. Kolari, T. Finin, and A. Joshi. Svms for the blogosphere: Blog identification and splog detection. In *AAAI Spring Symposium on Computational Approaches to Analysing Weblogs*, 2006.

[12] A. Kolcz, X Sun, and J Kalita. Efficient handling of high-dimensional feature spaces by randomized classifier ensembles. In *SIGKDD*, 2002.

[13] J. Kolter and M. Maloof. Learning to detect malicious executables in the wild. In *SIGKDD*, 2004.

[14] A. Maguitman, F. Menczer, H. Roinestad, and A. Vespignani. Algorithmic detection of semantic similarity. In *WWW*, 2005.

[15] M. Bailey, J. Oberheide, J. Andersen, Z. M.Mao, F. ahanian, and J. Nazario. Automated classification and analysis of internet malware. *RAID 2007, LNCS*, 4637:178–197, 2007.

[16] H. Oh, S. Myaeng, and M. Lee. A practical hypertext catergorization method using links and incrementally available class information. In *SIGIR*, 2000.

[17] J.R. Quinlan. C4.5. programs for machine learning. In *San Mateo, CA: Morgan Kaufmann*, 1993.

[18] D. K. S. Reddy and A. K. Pujari. N-gram analysis for computer virus detection. *J Comput Virol*, pages 231–239, 2006.

[19] M. Schultz, E. Eskin, and E. Zadok. Data mining methods for detection of new malicious executables. In *Proccedings of 2001 IEEE Symposium on Security and Privacy*, pages 38–49, 2001.

[20] A. Sung, J. Xu, P. Chavez, and S. Mukkamala. Static analyzer of vicious executables (save). In *Proceedings of the 20th Annual Computer Security Applications Conference*, 2004.

[21] G.J. Tesauro, J.O. Kephart, and G.B. Sorkin. Neural network for computer virus recognition. In *IEEE Expert, 11:5-6)*, 1996.

[22] I.W. Tsang, J.T. Kwok, and P.M. Cheung. Core vector machines: Fast svm training on very large data sets. *J. Mach. Learn. Res.*, 6:363–392, 2005.

[23] U. Bayer, A. Moser, C. Kruegel, and E. Kirda. Dynamic analysis of malicious code. *J Comput Virol*, 2:67–77, May 2006.

[24] J. Wang, P. Deng, Y. Fan, L. Jaw, and Y. Liu. Virus detection using data mining techniques. In *Proccedings of ICDM'03)*, 2003.

[25] X. Jiang and X. Zhu. vEye: behavioral footprinting for self-propagating worm detection and profiling. *Knowledge and Information System*, 18(2):231–262, 2009.

[26] Y. Yang, Seán Slattery, and R. Ghani. A study of approaches to hypertext categorization. *J. Intell. Inf. Syst.*, 18:219–241, March 2002.

[27] Y. Ye, D. Wang, T. Li, and D. Ye. IMDS: Intelligent malware detection system. In *SIGKDD*, 2007.

[28] H. Yu, J. Yang, and J. Han. Classifying large data sets using svms with hierarchical clusters. In *SIGKDD*, 2003.

[29] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Scholkopf. Learning with local and global consistency. In *Advances in Neural Information Processing Systems 16*, 2004.

[30] D. Zhou, S. Zhu, K. Yu, X. Song, B. L. Tseng, H. Zha, and C. Lee Giles. Learning multiple graphs for document recommendations. In *WWW*, 2008.

[31] S. Zhu, K. Yu, Y. Chi, and Y. Gong. Combining content and link for classification using matrix factorization. In *SIGIR*, 2007.