Ensemble Clustering for Internet Security Applications

Weiwei Zhuang, Yanfang Ye, Yong Chen, and Tao Li

***Abstract*—Due to their damage to Internet security, malware and phishing website detection has been the Internet security topics that are of great interests. Compared with malware attacks, phishing website fraud is a relatively new Internet crime. However, they share some common properties: 1) both malware samples and phishing websites are created at a rate of thousands per day driven by economic benefits; and 2) phishing websites represented by the term frequencies of the webpage content share similar characteristics with malware samples represented by the instruction frequencies of the program. Over the past few years, many clustering techniques have been employed for automatic malware and phishing website detection. In these techniques, the detection process is generally divided into two steps: 1) feature extraction, where representative features are extracted to capture the characteristics of the file samples or the websites; and 2) categorization, where intelligent techniques are used to automatically group the file samples or websites into different classes based on computational analysis of the feature representations. However, few have been applied in real industry products. In this paper, we develop an automatic categorization system to automatically group phishing websites or malware samples using a cluster ensemble by aggregating the clustering solutions that are generated by different base clustering algorithms. We propose a principled cluster ensemble framework to combine individual clustering solutions that are based on the consensus partition, which can not only be applied for malware categorization, but also for phishing website clustering. In addition, the domain knowledge in the form of sample-level/website-level constraints can be naturally incorporated into the ensemble framework. The case studies on large and real daily phishing websites and malware collection from the Kingsoft Internet Security Laboratory demonstrate the effectiveness and efficiency of our proposed method.**

***Index Terms*—Cluster ensemble, malware categorization, phishing website detection.**

I. Introduction

A. Malware Categorization and Phishing Website Detection

1) Malware Categorization: The proliferation of malware (such as virus, worms, Trojan Horses, spyware, backdoors, and rootkits) has presented a serious threat to the security of computer systems. Currently, the most significant line of defense against malware is Internet security software products, which mainly use a signature-based method to recognize threats in the clients. Given a collection of malware samples, these venders first categorize the samples into families so that samples in the same family share some common traits, and generate the common string(s) to detect variants of a family of malware samples.

2) Phishing Website Detection: Compared with malware attack, phishing website fraud is a relatively new Internet crime. Phishing is a form of online fraud, whereby perpetrators adopt social engineering schemes by sending e-mails, instant messages, or online advertising to allure users to phishing websites that impersonate trustworthy websites in order to trick individuals into revealing their sensitive information (e.g., financial accounts, passwords, personal identification numbers) which can then be used for profit [20]. To defend against phishing websites, security software products generally use blacklisting to filter against known websites. However, there is always a delay between website reporting and blacklist updating. Indeed, as lifetimes of phishing websites are reduced to hours from days, this method might be ineffective.

Although malware attack and phishing website fraud are two different forms of Internet security threat, they share several common properties. 1) Driven by economic benefits, both malware samples and phishing websites are increasing rapidly in creation frequency and sophistication. For example, the number of new phishing websites that are collected by the Antivirus Laboratory of Kingsoft is usually larger than 20 000 per day, and the number of new malware samples with various families collected by the Antivirus Laboratory of Kingsoft is usually larger than 10 000 per day. There is, thus, an urgent need of effective methods for automatic detection for these threats. 2) Though the phishing websites and the malware samples evolve constantly, most of their essence or the inherent structure is relatively stable. For example, a family of malware samples typically exhibit similar behavior profiles [4]. It has also been shown that phishing websites are not isolated from their targets but have strong relationships with them [24], which can be used as clues to cluster them into families and generate the signature for detection.

Over the past few years, many research efforts have been conducted on developing clustering techniques for automatic malware categorization [4], [6], [15], [17], [22], [39] and for phishing website detection and prevention [9], [20], [21]. In these systems, the detection process is generally divided into two steps: feature extraction and categorization. In the first step, various representative features (such as application programming interface (API) calls and instruction sequences for file

Manuscript received September 20, 2011; revised June 30, 2012; accepted September 5, 2012. Date of current version December 17, 2012. The work of W. Zhuang, Y. Ye, and Y. Chen was supported by the Guangdong Province Foundation under Grant 2008A090300017. The work of T. Li was supported in part by the U.S. National Science Foundation under Grant IIS-0546280, Grant DBI-0850203, Grant DMS-0915110, and Grant HRD-0833093. *(Corresponding author: T. Li.)* This paper was recommended by Associate Editor A. MIn Tjoa.

W. Zhuang is with the Department of Cognitive Science, Xiamen University, Xiamen 361005, China (e-mail: zhuangweiwei@gmail.com).

Y. Ye and Y. Chen are with the Internet Security R&D Center, Kingsoft Corporation, Zhuhai 519015, China (e-mail: yeyanfang@yahoo.com.cn).

T. Li is with the School of Computer Science, Florida International University, Miami, FL 33199 USA (e-mail: taoli@cs.fiu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMCC.2012.2222025

samples and term frequencies of webpage content for websites) are extracted to capture the characteristics of the file samples or the websites. In the second step, intelligent techniques are used to automatically categorize the file samples or the websites into different classes based on computational analysis of the feature representations. These techniques are varied in their use of feature representations and categorization methods. However, few have been applied in real industry products. In addition, clustering is an inherently difficult problem due to the lack of supervision information. Different clustering algorithms and even multiple trials of the same algorithm may produce different results due to random initializations and stochastic learning methods [36].

B. Contributions of the Paper

In this paper, we first observe that phishing websites represented by the term frequencies of the webpage content share similar characteristics with malware samples represented by the instruction frequencies of the program (see more details in Section IV). Based on this observation, we develop an automatic categorization system (ACS) to automatically group phishing websites or malware samples into families that share some common characteristics using a cluster ensemble by aggregating the clustering solutions that are generated by different base clustering algorithms.

To overcome the results instability and improve clustering performance, our ACS system uses a cluster ensemble to aggregate the clustering solutions that are generated by different algorithms. We develop base clustering algorithms to account for the characteristics of both phishing website and malware feature representations and propose a novel cluster ensemble framework to combine individual clustering solutions. We show that the domain knowledge in the form of website-level/sample-level constraints can be naturally incorporated into the ensemble framework. To the best of our knowledge, this is the first work of applying such cluster ensemble methods for both phishing website categorization and malware categorization. In short, our ACS system has the following major traits.

1) *Well-chosen feature representations:* Term frequency of the webpage content is used to represent websites, while instruction frequency is used for malware feature expression. These features well represent variants of phishing websites and malware families, respectively, and both can be efficiently extracted. More important, these two types of feature representations have similar underlying feature distributions on their corresponding datasets, which make it possible for us to propose a uniform framework which is based on clustering ensemble for both Internet security applications.
2) *Carefully designed base clusterings:* The choice of base clustering algorithms is largely dependent on the underlying feature distributions. To deal with the irregular and skewed distributions of term-frequency features as well as instruction-frequency features, we adopt both hierarchical clustering (HC) and K-medoids (KM) algorithms to generate base clusterings.
3) *A principled cluster ensemble scheme:* Our ACS system uses a cluster ensemble scheme to combine the clustering solutions of different algorithms. Our cluster ensemble scheme is a principled approach which is based on the consensus partition and is able to utilize the domain knowledge in the form of website-level/sample-level constraints.
4) *Incorporating domain knowledge:* In many cases, the domain knowledge and expertise of Internet security experts can greatly help improve the categorization results. Our ACS system offers a mechanism to incorporate the domain knowledge in the form of website-level/sample-level constraints (take malware categorization for example, some file samples are variants of a single malware; or some file samples belong to different malware types).

All these traits make our ACS system a practical solution for the application of Internet security including automatic phishing website detection and malware categorization. The case studies on large and real daily phishing website and malware collection from the Kingsoft Internet Security Laboratory demonstrate the effectiveness and efficiency of our proposed methods. As a result, our ACS system has already been incorporated into Kingsoft's Internet security software products. A preliminary conference version of this paper which focused on malware categorization is published in [45]. In this journal version, observing the commonality between malware categorization and phishing website detection, we tackle both problems via unified clustering ensemble framework.

C. Organization of the Paper

The rest of this paper is organized as follows. Section II presents the overview of our ACS system, and Section III discusses the related work. Section IV describes the feature extraction and representation of phishing websites as well as malware samples; Section V introduces the base clustering methods that we proposed to account for the characteristics of both phishing website and malware feature representations; Section VI presents the cluster ensemble framework that is used in our ACS system. In Section VII, using the daily data collection obtained from the Kingsoft Internet Security Laboratory, we systematically evaluate the effects and efficiency of our ACS system in comparison with other proposed clustering methods, as well as some of the popular Internet Security software such as Kaspersky and NOD32. Finally, Section VIII concludes this paper.

II. System Architecture

Fig. 1 shows the architecture of the ACS, and we briefly describe each component below.

1) *Term-frequency feature extractor:* For phishing website categorization, the ACS first uses the term-frequency feature extractor to extract the terms from the webpages of the collected phishing websites, and then transforms the data into term-frequency feature vectors. These vectors are stored in the database. The transaction data can also be easily converted to relational data if necessary.

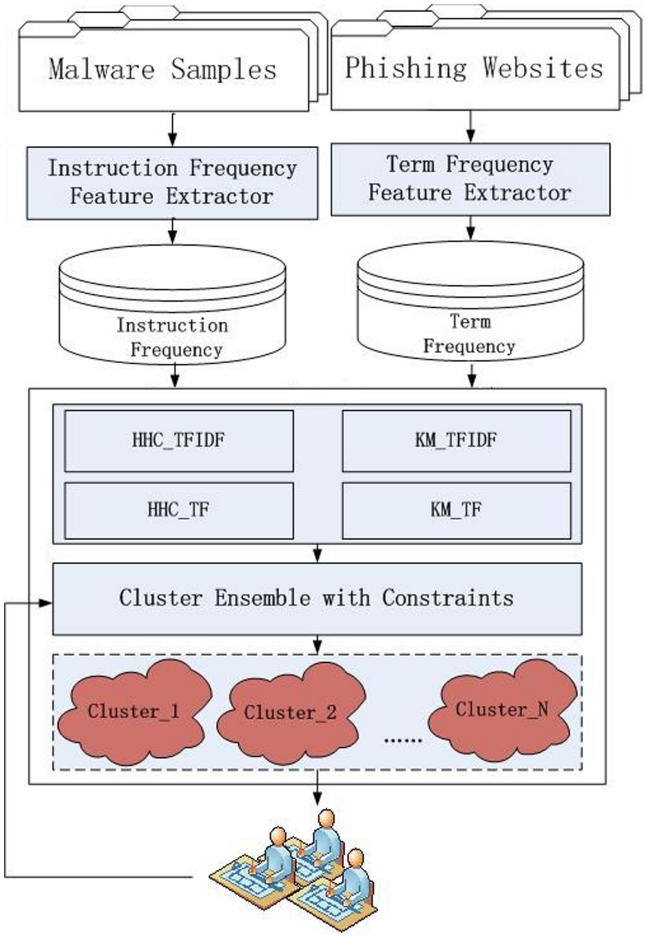

Fig. 1. Architecture of of the ACS.

2) *Instruction-frequency feature extractor:* For malware categorization, the ACS first uses the instruction-frequency feature extractor to extract the function-based instructions from the collected Portable Executable (PE) malware samples, converts the instructions to a group of 32-bit global IDs as the features of the data collection, and stores these features in the signature database. These integer vectors are then transformed to instruction frequencies and stored in the database. The transaction data can also be easily converted to relational data if necessary.
3) *Base clustering algorithms:* Base clustering solutions are generated by applying different clustering algorithms that are based on the feature representations. The HC algorithm and KM partitioned approach are applied on the term-frequency vectors or instruction-frequency vectors with the TF-IDF and TF weighting schemes [37], which are widely used for document representation in IR (information retrieval).
4) *Cluster ensemble with constraints:* Cluster ensemble is used to combine different base clusterings. The cluster ensemble is also able to utilize the domain knowledge in the form of website-level/sample-level constraints.
5) *Domain knowledge:* Our system provides a user-friendly mechanism to incorporate the expert knowledge and expertise of human experts. Internet security experts can look at the partitions and manually generate website-level/sample-level constraints. These constraints can be used to improve the categorization performance.

III. Related Work

A. Malware Categorization and Phishing Website Detection

1) Malware Feature Extraction: Features are the characterization of the behavior of a program under analysis. They are used as the input to data mining algorithms and can be derived from different levels of abstractions, including instruction level, API level, and cross-module level. There are generally three categories of feature extraction methods: dynamic, static, and hybrid. Dynamic analysis techniques observe the execution of the malware to derive features. The execution can be on a real or virtual processor. Well-known techniques include debugging and profiling. Example tools include Valgrind [30], QEMU [31], and strace. One advantage of dynamic feature extraction is that the environment- or configuration-dependent information has been resolved during the extraction, e.g., a variable whose value depends on the hardware, system configuration, or program input. One disadvantage of dynamic analysis is its limited coverage [29]. Static analysis techniques analyze the malware without running it. The target of analysis can be binary or source code. Static analysis has the advantage that it can explore all possible execution paths in the malware; therefore, it can be exhaustive in detecting malicious logic [10]. One disadvantage of static analysis is its inability to address certain situations due to undecidability, e.g., indirect control transfer through function pointers [28]. Hybrid analysis is an approach that combines static and dynamic analysis to gain the benefits of both [33]. In our study, built on our previous work [47], [48], we use the instruction-frequency feature extractor to extract the function-based instructions from the collected PE samples.

2) Malware Categorization: Various classification approaches including association classifiers, support vector machines, and Naive Bayes have been applied in malware detection [32], [35], [47]. HOLMES [13] detects malware families by combing frequent subgraph mining and concept analysis to synthesize discriminative specifications. Research efforts have been reported on combining different classification methods using different learning methods with possible different feature representations from malware detection [26], [46]. These classification methods require a large number of training samples to build the classification models. In recent years, there have been several initiatives in automatic malware categorization using clustering techniques [17]. Bayer *et al.* [4] used locality sensitive hashing and hierarchal clustering to efficiently group large datasets of malware samples into clusters. Lee and Mody [22] adopted KM clustering approach to categorize the malware samples. Several efforts have also been reported on computing the similarities between different malware samples using edit distance (ED) measure [15] or statistical tests [39].

3) Phishing Website Detection: Phishing website, a semantic attack which targets the user rather than the computer [2], is a relatively new significant security threat to the Internet in

comparison with malware [2]. Recently, many classification methods such as support vector machines and Naive Bayes have been used for antiphishing [1]. However, to date, to the best of our knowledge, there are only limited efforts that focus on phishing website clustering for phishing prevention [9], [20], [21]. Given an unknown webpage, Liu *et al.* [24], [25] proposed the following method for phishing detection: The method first finds the associated webpages with the given page, then mines the features (such as links relationship, ranking relationship, webpage text similarity, and webpage layout similarity relationship) between the given webpage and its associated webpages, and, finally, applies DBSCAN [11] clustering algorithm to decide if there is a cluster around the given webpage. If such cluster is found, the given webpage is then regarded as a phishing webpage; otherwise, it is identified as a legitimate webpage. Layton *et al.* [20], [21] proposed the following framework for phishing website clustering: It first extracts the bag-of-word representation from the source of the websites and then principal component analysis (PCA) for feature selection, and, finally, uses certain clustering algorithms (such as *k*-means, DBSCAN) for detection. For example, the experiments of [24] were performed based on 8745 phishing webpages and 1000 legitimate webpages, while Layton *et al.* [20] evaluated their proposed methods based on a dataset containing 24 403 websites.

We believe that the further progress can be made in clustering particular sets of malware samples or sets of phishing websites. In particular, existing clustering methods usually apply a specific clustering method on a feature representation. Different clustering methods have their own advantages and limitations in malware detection. In our study, we propose a principled cluster ensemble framework to integrate different clustering solutions.

B. Clustering Ensemble

Clustering ensemble refers to the process to obtain a single (consensus) and better-performing clustering solution from a number of different (input) clusterings for a particular dataset [36]. Many approaches have been developed to solve ensemble clustering problems over the past few years [3], [12], [16], [27], [40]. However, most of these methods are designed to combine partitional clustering methods, and few have been reported to combine both partitional and HC methods. In addition, they do not take advantage of the domain-related constraints. In our study, we use a cluster ensemble to aggregate the clustering solutions that are generated by both hierarchical and partitional clustering methods. Our ensemble framework is also able to incorporate the domain knowledge in the form of website-level/sample-level constraints.

IV. Feature Representation

A. Instruction Frequencies of Malware Samples

There are mainly two ways for feature extraction in malware analysis: static extraction and dynamic extraction. Dynamic feature extraction can well present the behaviors of malware files and especially perform well in analyzing packed malware [4], [22]. However, it has limited coverage: Only executable files can be executed or simulated. Actually, from the daily data collection of the Kingsoft Internet Security Laboratory, more than 60% of malware samples are dynamic link library files, which cannot be dynamically analyzed. In addition, dynamic feature extraction is time consuming. Therefore, in our study, we choose static feature extraction methods for malware representation. If a PE file is previously encrypt or compressed by a third-party binary compress tool such as UPX and ASPack Shell or embedded a homemade packer, it needs to be decrypt or decompressed first. We use the dissembler K32Dasm which was developed by the Kingsoft Internet Security Laboratory to dissemble the PE code and output the file of decrypt or unpacked format as the input for feature extraction. In this paper, we use the instruction frequencies for malware representation. The extraction and transformation processes are shown in Fig. 2. Comparing with other static features [39], such as construction phylogeny tree, control flow graph, Windows API calls, or arbitrary binaries, the instruction frequencies and function-based instruction sequences for malware representation have great ability to represent variants of a malware family, high coverage rate of malware samples, good semantic implications, and high efficiency for feature extraction [45].

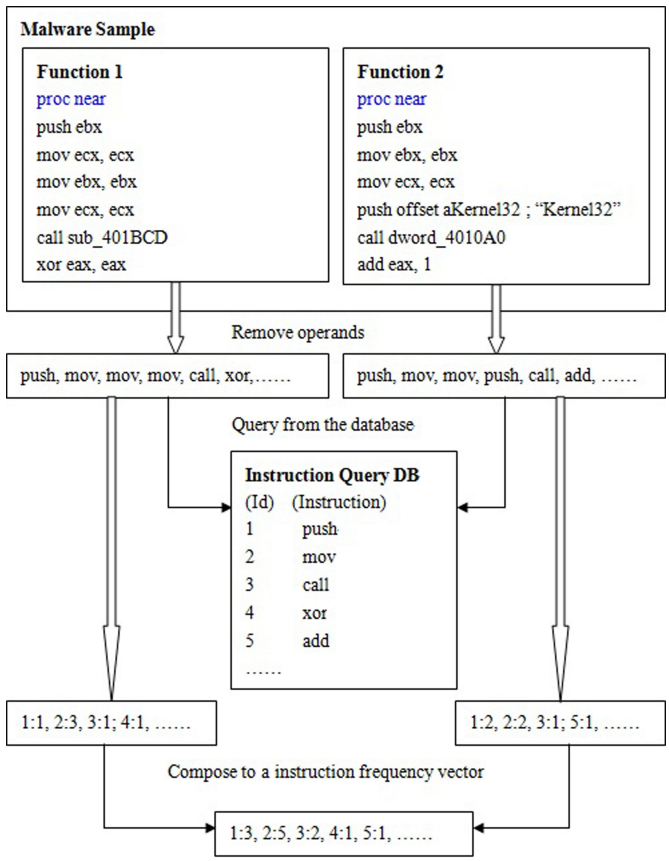

Fig. 2. Malware feature extraction and transformation processes of the ACS.

B. Term Frequencies of Phishing Websites

There are several feature extraction methods for phishing website representation: URL of the website [8], user interface

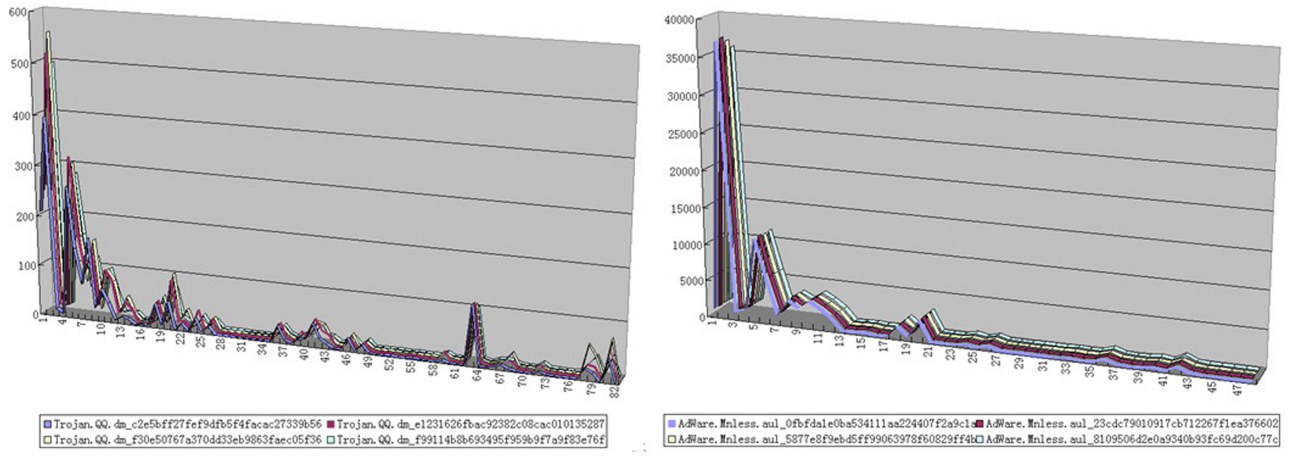

Fig. 3. Shapes of instruction-frequency patterns are shared by the same malware family and differ between different families.

Fig. 4. Two phishing websites of the same family share similar term features.

[43], associated webpages of the website [24], webpage block, layout, and overall style [25], terms of given webpage with the TF-IDF scores [9], [49], etc. Considering the expression ability of the website and the complexity for the categorization inputs, in this paper, we extract the term frequencies from the webpages of their corresponding websites. We first extract the terms from the "Title," "Keywords," "Description," "Copyright," and "Alt" of the webpages. The description of the extraction is illustrated as follows.

1) *Title:* extracting the content from the title tag of the webpage, i.e., the content between "<TITLE>. . .</TITLE>."
2) *Keywords:* extracting the keyword information of the website from the meta tag of the webpage, i.e., the content between "<META name=description content=. . .>."
3) *Description:* extracting the description information of the website from the meta tag of the webpage, i.e., the content between "<META name=keywords content=. . .>."
4) *Copyright:* extracting the copyright information of the website from the meta tag of the webpage, i.e., the content between "<META name=copyright content=. . .>."
5) *Alt:* extracting the text from the Alt tag of the webpage, i.e., the content between "<IMG alt=. . .>."

C. Characteristics of the Feature Representation

Note that phishing websites represented by the term frequencies of the webpage content share similar characteristics with malware samples represented by the instruction frequencies.

First, the feature representation is representative and can well group the instances of the same cluster. It has been observed in practice that malware samples in the same family or derived from the same source code share similar shapes of instruction-frequency patterns. Fig. 3 illustrates that the shapes of instruction-frequency patterns are similar for the same malware family, and they are different for different malware families. For websites, the extracted terms can well summarize the content of the full webpages, while eliminating a large amount of "redundant" information. As shown in Fig. 4, the two websites "http://www.nanhang10.tk/" and "http://www.zgnh-air.com" belong to the same family (sharing similar term features), which both masquerade as the real China Southern

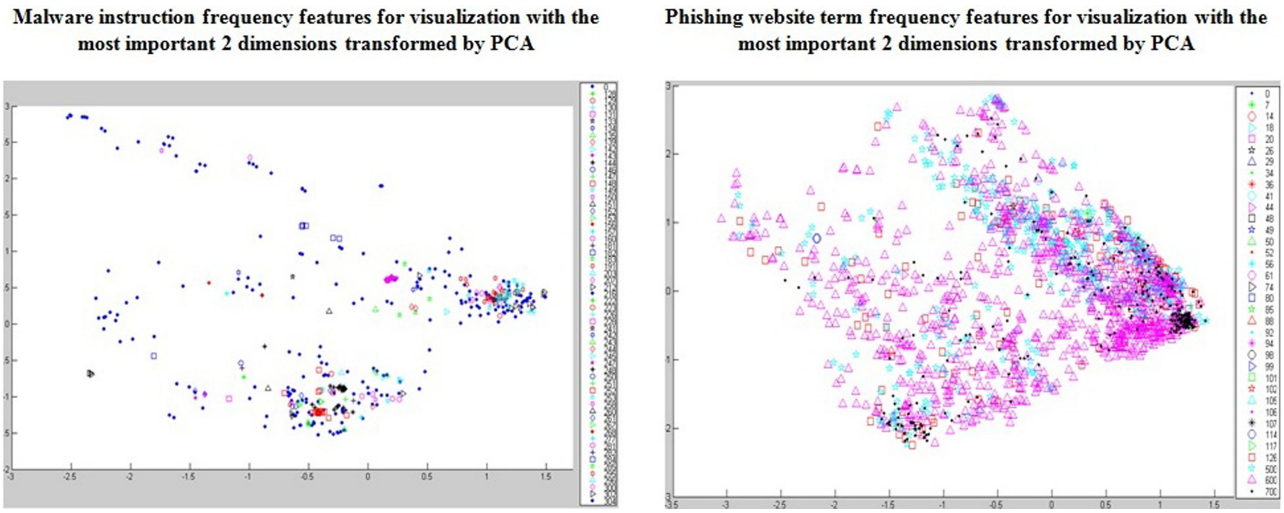

Fig. 5. Feature distributions after PCA transformation.

Airline to trick people into ordering the flight tickets and remitting money to the perpetrators.

Second, the term frequencies of the webpage content and the instruction frequencies of file samples have similar distribution patterns. Fig. 5 shows the distribution of term frequency on a set of 2004 phishing websites with 3038 dimensions as well as instruction frequency on a sample dataset with 1434 malware samples with 1222 dimensions. These two features with TF-IDF scheme [37] have been extracted, and PCA is performed to extract the first two important dimensions for visualization. As shown in Fig. 5, the distributions of phishing websites and malware samples are typically skewed, irregular, and of varied densities.

V. Base Clusterings

In our application, a cluster is a collection of phishing websites or malicious files that share some common traits between them and are "dissimilar" to the phishing websites or malware samples belonging to other clusters. Hierarchical and partitioning clustering are two common types of clustering methods, and each of them has its own traits [44]. The HC method can deal with irregular dataset more robustly, while partitioning clustering like KM is efficient and can produce tighter clusters especially if the clusters are of globular shape.

The choice of clustering algorithms is largely dependent on the underlying feature distributions. Since the feature distributions of malware samples and phishing websites are complex (as shown in Fig. 5), in our study, both HC and KM algorithms will be applied to generate base clusterings.

A. Hiearchical Clustering Algorithm

Hierarchical algorithms can be categorized into two subcategories [38], [42]: agglomerative algorithms and divisive algorithms. Because of its lower computation cost, in our application, we utilize the agglomerative HC algorithm as the frame, starting with N singleton clusters, and successively merges the two nearest clusters until only one cluster remains. The outline of the adopted hierarchical clustering (HC in short) algorithm suitable for both phishing website and malware categorization is described in Algorithm 1.

Algorithm 1. The algorithm description of HC.

Input: The data set D
Output: The best K and data clusters

Set each data point as a singleton cluster;
for $K \leftarrow N-1$ **to** 1 **do**
  Merge two closest clusters $C1$ and $C2$ into new cluster C with —$C1$—+—$C2$— elements;
  Calculate the similarity from C to all other clusters and update the similarity matrix;
  Calculate the validity index;
  Compare and keep the best K and corresponding clusters until now ;
end
Return the best K and corresponding clusters.

Here, we adopt *cosine similarity* [44] to measure the similarity between two data points, because of its independent data length. The definition of *cosine similarity* measure is described as

$$D_{ij} = \cos\alpha = \frac{x_i^T x_j}{|x_i||x_j|} \tag{1}$$

where x_i or x_j represent the vectors of the two data points. There are a variety of ways to compute the similarity from C to all other clusters: complete linkage, single linkage and average linkage [44]. Complete linkage is strongly biased toward producing clusters with roughly equal diameters, and it can be severely distorted by moderate outliers. Single linkage sacrifices performance in the recovery of compact clusters in return for the ability to detect elongated and irregular clusters. Considering the characteristics of both term-frequency and instruction-frequency feature representations, average linkage is used in our application.

For validity index, we use the *Fukuyama–Sugeno index (FS)* [14] to measure the quality of the clustering results. *FS* evaluates the partition by exploiting the compactness within each cluster and the distances between the cluster representatives. It

Algorithm 2. The algorithm description of KM.

Input: N points in d-dimensional space, number of clusters k
Output: k clusters

Randomly choose k cluster medoids;

repeat
  Assign each points to the nearest cluster;
  Update the cluster medoid by the calculation of validity index;
until *the medoids do not change*;

is defined as

$$FS = \sum_{i=1}^{N} \sum_{j=1}^{nc} u_{ij}^{m} (\|x_i - v_j\|_A^2 - \|v_j - v\|_A^2) \tag{2}$$

where v_j is the center [19] of cluster C_j, v is the center of the whole data collection, and A is an $n \times n$ positive definite, symmetric matrix where n is the feature dimension. It is clear that for compact and well-separated clusters, we expect small values for *FS*.

B. K-Medoids Clustering Approach

Another well-known clustering algorithm for categorization is squared error-based partitioning clustering, such as K-means [18] and KM [19], which assigns a set of data points into clusters using an iterative relocation technique [44]. A cluster is represented by one of its real data point (called medoids) or by the mean of its data points (called centroid) in KM and K-means methods, respectively. They are very simple, but effective and widely used in many scientific and industrial applications. Considering that the distributions of phishing websites and malware samples are typically skewed, irregular, and of densities, in order to well deal with the outlier problem, we use KM instead of K-means for categorization. The algorithm procedure for KM is described in Algorithm 2.

For the KM clustering algorithm, we use the same data point distance measure and validity index calculation methods as the aforementioned HC algorithm.

VI. Cluster Ensemble

A. Introduction

Clustering algorithms are valuable tools for malware categorization. However, clustering is an inherently difficult problem due to the lack of supervision information. Different clustering algorithms and even multiple trials of the same algorithm may produce different results due to random initializations and stochastic learning methods [36], [40]. In our study, we use a cluster ensemble to aggregate the clustering solutions that are generated by different both hierarchical and partitional clustering algorithms. We also show that the domain knowledge in the form of website-level/sample-level constraints can be naturally incorporated into the cluster ensemble. To the best of our knowledge, this is the first work of applying such cluster ensemble methods for Internet security including phishing website and malware categorizations.

B. Formulation

Formally, let $X = \{x_1, x_2, \ldots, x_n\}$ be a set of n data points (phishing websites or malware samples). Suppose that we are given a set of T clusterings (or partitioning) $\mathcal{P} = \{P^1, P^2, \ldots, P^T\}$ of the data points in X. Each partition $P^t (t = 1, \ldots, T)$ consists of a set of clusters $C^t = \{C_1^t, C_2^t, \ldots, C_{K_t}^t\}$, where K_t is the number of clusters for partition P^t and $X = \bigcup_{\ell=1}^{K} C_\ell^t$. Note that the number of clusters K could be different for different clusterings.

We define the *connectivity* matrix $M(P^t)$ for the partition P^t as

$$M_{ij}(P^t) = \begin{cases} 1 & \text{if } x_i \text{ and } x_j \text{ belong to the same cluster in } C^t \\ 0 & \text{Otherwise.} \end{cases} \tag{3}$$

Using the connectivity matrix, the distance between two partitions P^a, P^b can be defined as follows [16], [23]:

$$\begin{aligned} d(P^a, P^b) &= \sum_{i,j=1}^{n} d_{ij}(P^a, P^b) \\ &= \sum_{i,j=1}^{n} |M_{ij}(P^a) - M_{ij}(P^b)| \\ &= \sum_{i,j=1}^{n} [M_{ij}(P^a) - M_{ij}(P^b)]^2. \end{aligned}$$

Note that $|M_{ij}(P^a) - M_{ij}(P^b)| = 0$ or 1.

A general way for cluster ensemble is to find a *consensus* partition $\mathcal{P}^*$ which is the closest to all the given partitions:

$$\begin{aligned} \min_{P^*} J &= \frac{1}{T} \sum_{t=1}^{T} d(P^t, P^*) \\ &= \frac{1}{T} \sum_{t=1}^{T} \sum_{i,j=1}^{n} [M_{ij}(P^t) - M_{ij}(P^*)]^2. \end{aligned} \tag{4}$$

Since J is convex in $M(P^*)$, by setting $\nabla_{M(P^*)} J = 0$, we can easily show that the partition P^* that minimizes (4) is the consensus (average) association: the ijth entry of its connectivity matrix is

$$\widetilde{M}_{ij} = \frac{1}{T} \sum_{t=1}^{T} M_{ij}(P^t). \tag{5}$$

Proposition 6.1: The partition P^* that minimizes (4) is the consensus (average) association $\widetilde{M}_{ij}$.

In our application, we construct four base categorizers using the algorithms that are described in Section V. 1) Two clusterings are obtained by applying HC on the term-frequency vectors or instruction-frequency vectors with TF-IDF and TF weighting schemes (denoted by HC_TFIDF and HC_TF); and 2) two clusterings by applying KM on the term-frequency vectors or instruction-frequency vectors with TF-IDF and TF weighting

schemes with two different number of clusters: one is generated by HC_TFIDF, while the other is generated by HC_TF.

Based on Proposition 6.1, we could derive the final clustering from the consensus association $\widetilde{M}_{ij}$. The ijth entry of $\widetilde{M}_{ij}$ represents the number of times that data point i and j have cooccurred in a cluster. We could then use the following simple strategy to generate the final clustering. 1) For each data point pair, (i, j), such that $\widetilde{M}_{ij}$ is greater than a given threshold (in our application, the threshold is $0.5 \times 4 = 2$), assign the data points to the same cluster. If the data points were previously assigned to two different clusters, then merge these clusters into one. 2) For each remaining data point not included in any cluster, form a single element cluster. Note that we do not need to specify the number of clusters.

C. Incorporating Sample-Level Constraints

We also show that the domain knowledge in the form of website-level/sample-level constraints can be naturally incorporated into the cluster ensemble. In this scenario, in addition to t partitions, we are also given two sets of pairwise constraints: 1) must-link constraints

$$A = \{(x_{i1}, x_{j1}), \ldots, (x_{ia}, x_{ja})\}, a = |A|$$

where each pair of points are considered similar and should be clustered into the same cluster; and 2) cannot-link constraints

$$B = \{(x_{p1}, x_{q1}), \ldots, (x_{pb}, x_{pb})\}, b = |B|$$

where each pair of points are considered dissimilar, and they cannot be clustered into the same clusters. Such constraints have been widely used in *semisupervised clustering* [5]; however, few research efforts have been reported on incorporating constraints for cluster ensemble [41].

To incorporate the constraints in $\mathcal{M}$ and $\mathcal{C}$ into cluster ensemble, we need to solve the following problem:

$$\begin{aligned} \min_{P^*} J &= \frac{1}{T} \sum_{t=1}^{T} \sum_{i,j=1}^{n} [M_{ij}(P^t) - M_{ij}(P^*)]^2 \\ \text{s.t.} \quad & M_{ij}(P^*) = 1, \quad \text{if } (x_i, x_j) \in A \\ & M_{ij}(P^*) = 0, \quad \text{if } (x_i, x_j) \in B. \end{aligned} \tag{6}$$

Equation (6) is a convex optimization problem with linear constraints. Let $C = A \bigcup B$ be the set of all constraints; then $c = |C| = |A| + |B|$. We can represent C as $C = \{(x_{i1}, x_{j1}, b_1), \ldots, (x_{ic}, x_{jc}, b_c)\}$, where $b_s = 1$ if $(x_{is}, x_{js}) \in A$, and $b_s = 0$ if $(x_{is}, x_{js}) \in B, s = 1 \ldots c$. We can then rewrite (6) as

$$\begin{aligned} \min_{P^*} J &= \frac{1}{T} \sum_{t=1}^{T} \sum_{i,j=1}^{n} [M_{ij}(P^t) - M_{ij}(P^*)]^2 \\ \text{s.t.} \quad & (\mathbf{e}_{i_s}) M(P^*) \mathbf{e}_{j_s} = b_s, s = 1, 2, \ldots, c \end{aligned} \tag{7}$$

where $\mathbf{e}_{i_s} \in \mathbb{R}^{n \times 1}$ is an indicator vector with only the i_sth element being 1 and all other elements being 0. Now, we introduce a set of Lagrangian multipliers $\{\alpha_i\}_{i=1}^{c}$ and construct the Lagrangian for problem (7) as

$$\mathcal{L} = J + \sum\nolimits_s \alpha_s \left((\mathbf{e}_{i_s})^T M(P^*) \mathbf{e}_{j_s} - b_s \right). \tag{8}$$

Note that $(\mathbf{e}_{i_s})^T M(P^*) \mathbf{e}_{j_s} = M_{i_s j_s}(P^*)$. Hence, we can show that the solution to problem (7) is

$$M_{i_s j_s}(P^*) = \begin{cases} \dfrac{1}{T} \displaystyle\sum_{t=1}^{T} M_{ij}(P^t) & \text{if } (i_s, j_s) \text{ is not in } C \\ b_s & \text{otherwise.} \end{cases} \tag{9}$$

In other words, the solutions for regular elements in $\widetilde{M}_{ij}$ do not change and for constrained elements, according to (9), we need to set the corresponding entries of the consensus association $\widetilde{M}_{ij}$ to be the exact values based on their constraints.

VII. Experimental Results and Analysis

In this section, we conduct two sets of experimental studies using our data collection obtained from the Kingsoft Internet Security Laboratory to evaluate the categorization methods that we proposed in this paper. 1) In the first set of experiments, on the basis of term-frequency features of the phishing webpages, we evaluate our proposed methods for phishing website categorization. 2) In the second part of experiments, resting on the analysis of instruction-frequency features extracted from the malware samples, we evaluate our proposed cluster ensemble for malware categorization. In this paper, we measure the categorization performance of different algorithms using Macro-F1 and Micro-F1 measures, which emphasize the performance of the system on rare and common categories, respectively [34]. All the experimental studies are conducted under the environment of Windows XP operating system plus Intel P4 1.83 GHz CPU and 2 GB of RAM.

A. Evaluation of the Proposed Cluster Ensemble for Phishing Website Categorization

In this set of experiments, we 1) first evaluate the effectiveness of phishing website categorization results of our proposed cluster ensemble, especially with website-level constraints; 2) and then compare our ACS system for phishing website categorization with some of the popular anti-phishing tools, such as Kaspersky Anti-Phishing, Netcraft, etc.

1) Evaluation of Cluster Ensemble With Constraints for Phishing Website Categorization: Using daily phishing websites and their corresponding webpages' collection obtained from the Kingsoft Internet Security Laboratory from June 10, 2012, to June 16, 2012, and resting on the term-frequency features extracted from the webpages, we construct the cluster ensemble using four base clusterings: HC_TFIDF and HC_TF as described in Section VI-B, and two KM categorizers with two different Ks. From Table I, we observe that the phishing website categorization results of the cluster ensemble outperform each individual algorithm.

It should be pointed out that in some cases, categorizing a phishing website to a certain family is still the prerogative of Internet security experts. For example, though some of the

TABLE I
EVALUATION OF THE PHISHING WEBSITE CATEGORIZATION RESULTS OF CLUSTERING ENSEMBLE

Date	Num	D	F	Alg	*Macro*	*Micro*
2012-06-10	10014	13424	131	HC_TFIDF	0.6719	0.6720
				HC_TF	0.7233	0.7613
				KM_TFIDF	0.6845	0.6931
				KM_TF	0.6526	0.6719
				NCE	0.8102	0.8415
				CE	0.8611	0.8853
2012-06-11	6351	7158	75	HC_TFIDF	0.7045	0.7136
				HC_TF	0.6845	0.7244
				KM_TFIDF	0.6884	0.7301
				KM_TF	0.7354	0.7287
				NCE	0.8047	0.8293
				CE	0.8545	0.8761
2012-06-12	7546	9735	96	HC_TFIDF	0.6835	0.7256
				HC_TF	0.7341	0.7295
				KM_TFIDF	0.7517	0.7268
				KM_TF	0.7030	0.7655
				NCE	0.8490	0.8644
				CE	0.8812	0.9102
2012-06-13	9415	10048	83	HC_TFIDF	0.6742	0.7153
				HC_TF	0.7305	0.7218
				KM_TFIDF	0.7125	0.7351
				KM_TF	0.6643	0.7033
				NCE	0.8145	0.8369
				CE	0.8588	0.8751
2012-06-14	4553	6274	70	HC_TFIDF	0.6952	0.6893
				HC_TF	0.7245	0.7568
				KM_TFIDF	0.6340	0.6438
				KM_TF	0.6402	0.6973
				NCE	0.8097	0.8455
				CE	0.8601	0.8882
2012-06-15	12053	15046	135	HC_TFIDF	0.7255	0.7249
				HC_TF	0.6870	0.7324
				KM_TFIDF	0.6865	0.7321
				KM_TF	0.7502	0.7403
				NCE	0.8496	0.8713
				CE	0.8859	0.9125
2012-06-16	9204	8842	87	HC_TFIDF	0.6841	0.7238
				HC_TF	0.7155	0.7202
				KM_TFIDF	0.7525	0.7236
				KM_TF	0.7138	0.7693
				NCE	0.8425	0.8632
				CE	0.8812	0.9079

Remark: Based on term frequency, the categorization results of different categorizers are performed on the real daily phishing website collection from June 10, 2012, to June 16, 2012. "Num"—the total number of the phishing websites, "D"—Dimensions of the dataset, "F"—the real phishing website families, "NCE"—cluster ensemble without constraints, "CE"—cluster ensemble with constraints, "Macro"—Macro-F1 measure, and "Micro"—Micro-F1 measure.

phishing websites are prize-winning fraud websites and share similar shape of term-frequency patterns and, thus, may be categorized to the same family, according to their exact intents, they should be divided into different families. On the contrary, there are some metamorphic phishing websites, like selling counterfeit medicine fraud, which may differ from term representations, but they are in the same family. In such cases, if we can add some website-level constraints, the categorization results will be improved. Our categorization system (ACS) provides a user-friendly mechanism to incorporate the expert knowledge and expertise of human experts. The cluster ensemble scheme of our categorization system not only combines the clustering results of individual categorizers, but also incorporates the website-level constraints provided by the human analysts. According to the

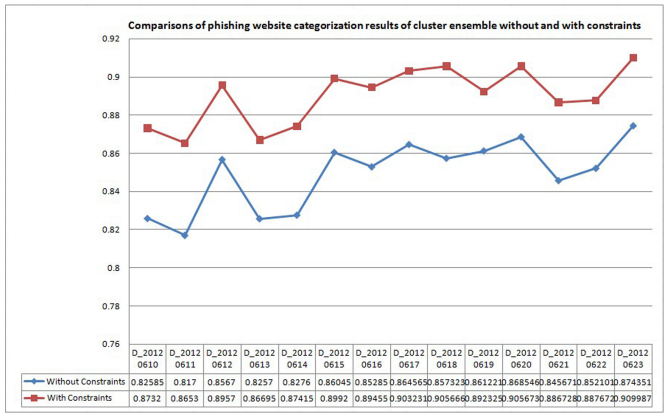

Fig. 6. Comparisons of phishing website categorization results of cluster ensembles without and with constraints.

expertise of the Internet security experts, the ACS now totally gets 3693 pairs of must-link constraints and 4857 cannot-link constraints.

To further demonstrate the advantage of incorporating website-level constraints, we use the real daily phishing websites and their corresponding webpages collection for two weeks (from June 10, 2012, to June 23, 2012) to compare the categorization results of cluster ensemble without constraints and with constraints. Experimental results in Fig. 6 (for comparison purpose, we use the average of the Macro-F1 and Micro-F1 values to evaluate the categorization results) clearly show that ensemble with constraints outperforms the one without constraints.

2) Comparisons With Different Anti-Phishing Tools for Phishing Website Categorization: Though there are some of the popular antiphishing tools for phishing website prevention, such as Kaspersky Anti-Phishing (Kasp), McAfee Anti-Phishing Tool SiteAdvisor (SiteAd), Netcraft which is the plug-in of Firefox Internet browser, and Rising Anti-Phishing tool, based on the same testing phishing webpages described in the aforementioned section, Table II shows that these antiphishing tools just have limited detection ability, especially for Chinese phishing website detection, and none of them have the functionalities for phishing website categorization.

In addition, we also evaluate the efficiency of our ACS system for phishing website categorization. It just takes minutes to finish the categorization tasks that are based on our daily collected data, for example, categorizing 10 014 phishing websites by our ACS system including feature extraction needs 3 min and 55 s, while categorizing 6351 needs 2 min and 5 s.

B. Evaluation of the Proposed Cluster Ensemble for Malware Categorization

In this section, based on the daily new malware sample collection obtained from the Kingsoft Internet Security Laboratory, we first 1) evaluate the effectiveness of malware categorization results of our proposed cluster ensemble compared with single-base clustering algorithms, especially with sample-level constraints; and 2) then compare our ACS system for malware categorization with some of the popular antimalware

TABLE II
CATEGORIZATION RESULTS OF DIFFERENT ANTIPHISHING TOOLS ON THE DAILY PHISHING WEBPAGE COLLECTION

Date	AP.	Detected	Families	$MacroF1$	$MicroF1$
2012-06-10	Kasp	304	—-	—-	—-
	SiteAd	985	—-	—-	—-
	NetCraft	3,402	—-	—-	—-
	Rising	9,441	—-	—-	—-
	ACS	10,014	131	0.8611	0.8853
2012-06-11	Kasp	164	—-	—-	—-
	SiteAd	759	—-	—-	—-
	NetCraft	2,024	—-	—-	—-
	Rising	5,837	—-	—-	—-
	ACS	6,351	75	0.8545	0.8761
2012-06-12	Kasp	179	—-	—-	—-
	SiteAd	813	—-	—-	—-
	NetCraft	2,904	—-	—-	—-
	Rising	6,954	—-	—-	—-
	ACS	7,546	96	0.8812	0.9102
2012-06-13	Kasp	253	—-	—-	—-
	SiteAd	942	—-	—-	—-
	NetCraft	3,526	—-	—-	—-
	Rising	8,895	—-	—-	—-
	ACS	9,415	83	0.8588	0.8751
2012-06-14	Kasp	118	—-	—-	—-
	SiteAd	524	—-	—-	—-
	NetCraft	1,533	—-	—-	—-
	Rising	3,982	—-	—-	—-
	ACS	4,553	70	0.8601	0.8882
2012-06-15	Kasp	336	—-	—-	—-
	SiteAd	1,042	—-	—-	—-
	NetCraft	3,701	—-	—-	—-
	Rising	11,364	—-	—-	—-
	ACS	12,053	135	0.8859	0.9125
2012-06-16	Kasp	233	—-	—-	—-
	SiteAd	915	—-	—-	—-
	NetCraft	3,347	—-	—-	—-
	Rising	8,749	—-	—-	—-
	ACS	9,204	87	0.8812	0.9079

TABLE III
EVALUATION OF THE MALWARE CATEGORIZATION RESULTS OF CLUSTERING ENSEMBLE

Date	Num	D	F	Alg	$Macro$	$Micro$
2012-06-10	3025	1203	185	HC_TFIDF	0.7246	0.8015
				HC_TF	0.7208	0.7754
				KM_TFIDF	0.7013	0.8025
				KM_TF	0.6016	0.7666
				NCE	0.8713	0.8854
				CE	0.9012	0.9118
2012-06-11	3538	1521	297	HC_TFIDF	0.7423	0.7767
				HC_TF	0.6855	0.7287
				KM_TFIDF	0.7287	0.7798
				KM_TF	0.7288	0.7803
				NCE	0.8689	0.8793
				CE	0.9001	0.9043
2012-06-12	2928	1096	178	HC_TFIDF	0.7743	0.7108
				HC_TF	0.7509	0.7543
				KM_TFIDF	0.6643	0.7237
				KM_TF	0.6765	0.7788
				NCE	0.8761	0.8884
				CE	0.8972	0.9058
2012-06-13	4532	2256	325	HC_TFIDF	0.7312	0.8009
				HC_TF	0.7195	0.7699
				KM_TFIDF	0.7234	0.8101
				KM_TF	0.6267	0.7721
				NCE	0.8673	0.8785
				CE	0.9073	0.9014
2012-06-14	3887	1859	253	HC_TFIDF	0.7513	0.7802
				HC_TF	0.6902	0.7366
				KM_TFIDF	0.7303	0.7805
				KM_TF	0.7322	0.7795
				NCE	0.8687	0.8753
				CE	0.8966	0.9095
2012-06-15	3749	2058	276	HC_TFIDF	0.7832	0.7239
				HC_TF	0.7672	0.7613
				KM_TFIDF	0.6702	0.7312
				KM_TF	0.6809	0.7858
				NCE	0.8765	0.8834
				CE	0.9012	0.9113
2012-06-16	4012	2561	321	HC_TFIDF	0.7714	0.7932
				HC_TF	0.7002	0.7533
				KM_TFIDF	0.7529	0.7967
				KM_TF	0.7552	0.7901
				NCE	0.8915	0.9029
				CE	0.9032	0.9126

Remark: Based on instruction frequency, the categorization results of different categorizers are performed on the real daily new malware collection from June 10, 2012, to June 16, 2012. "Num"—the total number of the malware samples, "D"—dimensions of the dataset, "F"—the real malware families, "NCE"—cluster ensemble without constraints, "CE"—cluster ensemble with constraints, "Macro"—Macro-F1 measure, and "Micro"—Micro-F1 measure.

software products such as Norton AntiVirus, Bitdefender, MaAfee VirusScan, and Kaspersky Anti-Virus.

1) Evaluation of Cluster Ensemble With Constraints for Malware Categorization: In this set of experiments, we evaluate the effectiveness of malware categorization results of our proposed cluster ensemble, especially with sample-level constraints. Using the daily new malware sample collection obtained from the Kingsoft Internet Security Laboratory from every 9:00 A.M. to 12:00 noon from June 10, 2012, to June 16, 2012 and resting on the instruction-frequency features that are extracted from the malware samples, we construct the cluster ensemble using four base clusterings: HC_TFIDF and HC_TF as described in Section VI-B, and two KM categorizers with two different Ks. From Table III, we observe that the malware categorization results of the cluster ensemble outperform each individual algorithm.

It should be pointed out that in many cases, categorizing a malware sample to a certain family is still the prerogative of Internet security experts. For example, as shown in Fig. 7, though some of the malware files complied by Delphi compiler or E-language compiler which uses Chinese for program development share similar shape of instruction-frequency patterns and, thus, may be categorized to a same family, according to their intents and behaviors, they should be divided into different families. On the contrary, there are some metamorphic malware samples, like "Trojan.Swizzors," which may differ from static feature representations, but they are in the same family. In such cases, if we can add some sample-level constraints, the categorization results will be improved. Our malware categorization system (ACS) provides a user-friendly mechanism to incorporate the expert knowledge and expertise of human experts. The cluster ensemble scheme of our malware categorization system not only combines the clustering results of individual categorizers, but also incorporates the sample-level constraints provided by the human analysts. According to the expertise of the malware analysts, the ACS now totally gets 4025 pairs of must-link constraints and 3958 pairs of cannot-link constraints.

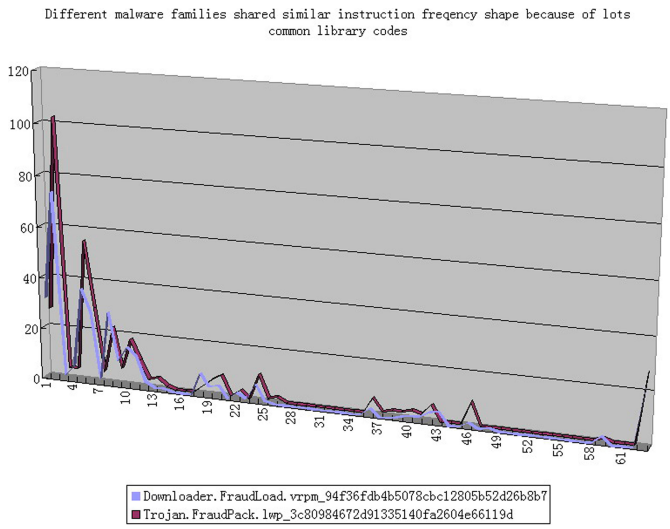

Fig. 7. Example of sample-level inequivalence constraints.

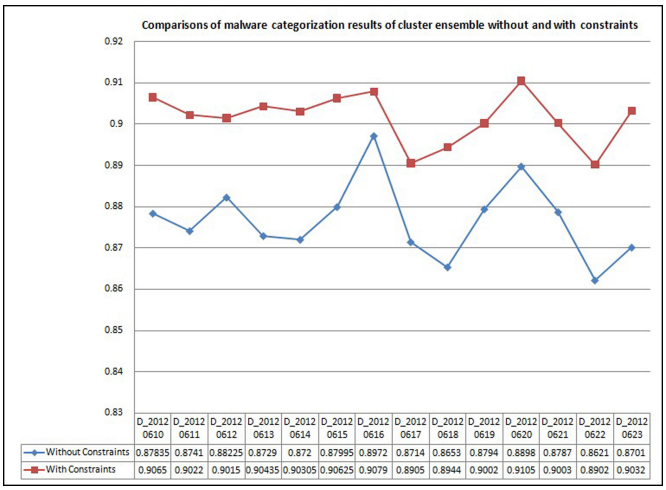

Fig. 8. Comparisons of malware categorization results of cluster ensembles without and with constraints.

To further demonstrate the advantage of incorporating sample-level constraints, we use the real daily new malware collection for two weeks (from June 10, 2012, to June 23, 2012) which totally includes 58 438 malware samples to compare the categorization results of cluster ensemble without constraints and with constraints. Experimental results in Fig. 8 (for comparison purpose, we use the average of the Macro-F1 and Micro-F1 values to evaluate the categorization results) clearly show that ensemble with constraints outperforms the one without constraints.

2) Comparisons With Different AV Venders for Malware Categorization: In this section, we apply the ACS in real applications to evaluate its malware categorization effectiveness and efficiency of the daily data collection. We use the whole data collection for two weeks (from June 10, 2012, to June 23, 2012) which consists of 58 438 malware samples with 3256 families to compare the malware categorization effectiveness of the ACS with some of the popular AV products, such as Kaspersky(Kasp), NOD32, Mcafee, Bitdefender(BD), and Rising. For comparison purpose, we use all of the Anti-Virus scanners' newest versions of the base of signature on the same day (June 23, 2012). Table IV and Fig. 9 show that the malware categorization effectiveness of our ACS outperforms other popular AV products.

TABLE IV
CATEGORIZATION RESULTS OF DIFFERENT AV SOFTWARE ON THE WHOLE DATA COLLECTION OF 58 438 MALWARE SAMPLES

AV.	Detected	Families	$MacroF1$	$MicroF1$
Kasp	51,342	2,657	0.7323	0.7477
Nod32	48,457	2,456	0.6523	0.6987
Mcafee	46,982	2,249	0.5867	0.6348
BD	54,005	2,989	0.7602	0.7823
Rising	50,557	2,523	0.5846	0.6398
ACS	56,206	3,012	0.9025	0.9081

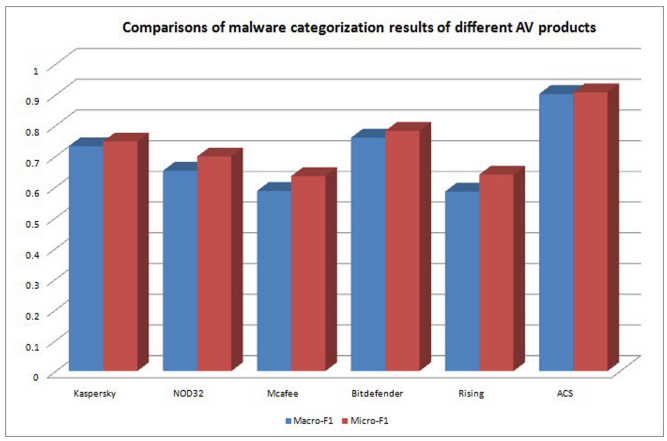

Fig. 9. Comparison of malware categorization results of different AV software on the whole data collection of 58 438 malware samples.

For robust evaluation, we track the malware categorization results of our ACS and AV software products above, based on 30 consecutive days (from May 25, 2012, to June 23, 2012) of new malware sample collection with a total number of 187 235. The real daily experiments demonstrate that the average of Macro-F1 and average of Micro-F1 of the ACS are higher than 0.88, while none of those five popular AV software are higher than 0.80. In addition, we also evaluate the efficiency of our ACS system: 1) Categorizing 3025 malware samples by our ACS system including feature extraction needs 50 s; and (2) the whole process of 58438 malware samples needs 12 min.

VIII. CONCLUSION

In this paper, we have developed an ACS which can not only be applied for phishing website categorization, but also for categorizing malware samples into families that share some common traits by an ensemble of different clustering solutions that are generated by different clustering methods. Empirical studies on large and real daily datasets that are collected by the Kingsoft Internet Security Laboratory illustrate that our ACS system performs well for real phishing website categorization as well as malware categorization applications.

There are many avenues for future works. First, we will explore various base clustering algorithms (e.g., recent probabilistic clustering methods and subspace clustering) with different feature representations. Second, we will extend our clustering ensemble framework for anomaly detection. Third, we will investigate new ways to represent domain knowledge and novel methods to incorporate domain knowledge into the detection process.

ACKNOWLEDGMENT

The authors would also like to thank the members of the Internet Security Laboratory at Kingsoft Corporation for their helpful discussions and suggestions.

REFERENCES

[1] S. Abu-Nimeh, D. Nappa, X. Wang, and S. Nair, "A comparison of machine learning techniques for phishing detection," in *Proc. APWG eCrime Res. Summit*, 2007, pp. 60–69.

[2] M. Aburrous, M. A. Hossain, K. Dahal, and F. Thabtah, "Predicting phishing websites using classification mining techniques with experimental case studies," in *Proc. 7th Int. Conf. Inf. Technol.*, 2010, pp. 176–181.

[3] J. Azimi and X. Fern, "Adaptive cluster ensemble selection," in *Proc. 21st Int. Joint Conf. Artif. Intell.*, San Francisco, CA, 2009, pp. 992–997.

[4] M. Bailey, J. Oberheide, J Andersen, Z. M. Mao, F. Jahanian, and J. Nazario, "Automated classification and analysis of internet malware," in *Recent Advances in Intrusion Detection*, (Lecture Notes in Computer Science vol. 4637). New York: Springer, 2007, pp. 178–197.

[5] S. Basu, I. Davidson, and K. L. Wagstaff, Eds., *Constrained Clustering: Advances in Algorithms, Theory, and Applications*. Boca Raton, FL: CRC Press, 2008.

[6] U. Bayer, P. M. Comparetti, C. Hlauschek, C. Kruegel, and E. Kirda, "Scalable, behavior-based malware clustering," in *Proc. 16th Annu. Netw. Distributed Secur. Symp.*, 2009.

[7] C. Herley and D. Florencio, "A profitless endeavor: Phishing as tragedy of the commons," in *Proc. New Secur. Paradigms Workshop*, 2008.

[8] N. Chou, R. Ledesma, Y. Teraguchi, D. Boneh, and J. C. Mitchell, "Client-side defense against web-based identity theft," in *Proc. 11th Annu. Network Distrib. Syst. Secur. Symp.*, 2004.

[9] R. Dazeley, J. L. Yearwood, B. H. Kang, and A. V. Kelarev, "Consensus clustering and supervised classification for profiling phishing emails in internet commerce security," in *Knowledge Management and Acquisition for Smart Systems and Service* (Lecture Notes in Computer Science, vol. 6232). New York, Springer-Verlag, 2010, pp. 235–246.

[10] Y. Elovici, A. Shabtai, R. Moskovitch, G. Tahan, and C. Glezer, "Applying machine learning techniques for detection of malicious code in network traffic," in *KI 2007: Advances in Artificial Intelligence* (Lecture Notes in Computer Science, vol. 4667). Berlin, Heidelberg: Springer-Verlag, 2007, pp. 44–50.

[11] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial database with noise," in *Proc. ACM Int. Conf. Knowl. Discovery Data Mining*, 1996, pp. 226–231.

[12] X. Fern and C. E. Brodley, "Solving cluster ensemble problems by bipartite graph partitioning," in *Proc. 21st Int. Conf. Mach. Learn.*, 2004, p. 36.

[13] M. Fredrikson, S. Jha, M. Christodorescu, R. Sailer, and X. Yan, "Synthesizing near-optimal malware specifications from suspicious behaviors," in *Proc. IEEE Symp. Secur. Priv.,* Washington, DC IEEE Computer Society, May 2010, pp. 45–60.

[14] Y. Fukuyama and M. Sugeno, "A new method of choosing the number of clusters for the fuzzy C-means method," in *Proc. 5th Fuzzy Syst. Sym.*, 1989, pp. 247–250.

[15] M. Gheorghescu, "An automated virus classification system," in *Proc. VIRUS BULLETIN CON.*, Oct. 2005.

[16] A. Gionis, H. Mannila, and P. Tsaparas, "Clustering aggregation," in *Proc. 21st Int. Conf. Data Eng.*, 2005, pp. 341–352.

[17] I. Gurrutxaga, O. Arbelaitz, J. M. Perez, J. Muguerza, J. I. Martin, and I. Perona, "Evaluation of Malware clustering based on its dynamic behaviour.," in *Proc. 7th Australas. Data Mining Conf.*, 2008, pp. 163–170.

[18] J. Hartigan and M. Wong, "Algorithm AS136: A k-means clustering algorithm," *J. Roy. Stat. Soc., Appl. Stat.*, vol. 28, pp. 100–108, 1979.

[19] L. Kaufman and P. J. Rousseeuw, "Partitioning around medoids (program PAM)," *Finding Groups in Data: An Introduction to Cluster Analysis*, 1990.

[20] R. Layton, S. Brown, and P. Watters, "Using differencing to increase distinctiveness for phishing website clustering," in *Proc. Symp. Workshops Ubiquitous, Autonom. Trusted Comput.*, 2009, pp. 488–492.

[21] R. Layton and P. Watters, "Determining provenance in phishing websites using automated conceptual analysis," in *Proc. eCrime Res. Summit*, 2009, pp. 1–7.

[22] T. Lee and J. J. Mody, "Behavioral classification," in *Proc. EICAR*, May 2006.

[23] T. Li and C. Ding, "Weighted Consensus Clustering," in *Proc. SIAM Int. Conf. Data Mining*, 2008, pp. 798–809.

[24] G. Liu, B. Qiu, and L. Wenyin, "Automatic detection of phishing target from phishing webpage," in *Proc. 20th Int. Conf. Pattern Recognit.*, 2010, pp. 4153–4156.

[25] W. Liu, X. Deng, G. Huang, and A. Y. Fu, "An antiphishing strategy based on visual similarity assessment," in *Proc. IEEE Internet Comput.*, Mar./Apr. 2006, pp. 58–65.

[26] E. Menahem, A. Shabtai, L. Rokach, and Y. Elovici, "Improving malware detection by applying multi-inducer ensemble," *J. Comput. Stat. Data Anal.*, vol. 53, no. 4, pp. 1483–1494, Feb. 2009.

[27] S. Monti, P. Tamayo, J. Mesirov, and T. Gloub, "Consensus clustering: A resampling-based method for class discovery and visualization of gene expression microarray data," *Mach. Learn. J.*, vol. 52, no. 1–2, pp. 91–118, 2003.

[28] A. Moser, C. Kruegel, and E. Kirda, "Limits of static analysis for malware detection," in *Proc. 23rd Annu. Computer Secur. Appl. Conf.*, 2007, pp. 421–430.

[29] A. Moser, C. Kruegel, and E. Kirda, "Exploring multiple execution paths for malware analysis," in *Proc. IEEE Symp. Secur. Privacy*, May 2007, pp. 231–245.

[30] N. Nethercote and J. Seward, "Valgrind: A framework for heavyweight dynamic binary instrumentation," presented at the ACM SIGPLAN 2007 Conf. Program. Lang. Des. Implementation (PLDI 2007), San Diego, CA, Jun. 2007.

[31] QEMU. (2012). [Online]. Available: http://www.qemu.org/

[32] K. Rieck, T. Holz, C. Willems, P. Dussel, and P. Laskov, "Learning and classification of malware behavior," in *Proc. 5th Conf. Detect. Intrusions Malware & Vulnerability Assessment*, 2008, pp. 108–125.

[33] P. Royal, M. Halpin, D. Dagon, R. Edmonds, and W. Lee, "PolyUnpack: Automating the hidden-code extraction of unpack-executing malware," in *Proc. 22nd Annu. Computer Sec. Appl. Conf.*, Miami Beach, FL, 2006.

[34] F. Sebastiani, "Text categorization," *ACM Comput. Surveys*, vol. 34, pp. 1–47, 2002.

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

[36] A. Strehl and J. Ghosh, "Cluster ensembles—A knowledge reuse framework for combining multiple partitions," *J. Mach. Learn. Res.*, vol. 3, pp. 583–617, Mar. 2003.

[37] K. Sugiyama, K. Hatano, M. Yoshikawa, and S. Uemura, "Refinement of TF-IDF schemes for web pages using their hyperlinked neighboring pages," in *Proc. 14th ACM Conf. Hypertext and Hypermedia*, Aug. 2003, pp. 198–207.

[38] S. Theodoridis and K. Koutroubas, *Pattern Recognition*. New York: Academic, 1999.

[39] R. Tian, L. M. Batten, and S. C. Versteeg, "Function length as a tool for malware classification," in *Proc. 3rd Int. Conf. Malicious Unwanted Software*, 2008, pp. 69–76.

[40] A. P. Topchy, A. K. Jain, and W. F. Punch, "Clustering ensembles: Models of consensus and weak partitions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 12, pp. 1866–1881, Dec. 2005.

[41] F. Wang, X. Wang, and T. Li, "Generalized cluster aggregation," in *Proc. 21st Int. Joint Conf. Artif. Intell.*, 2009, pp. 1279–1284.

[42] C. Williams, "A MCMC approach to hierarchical mixture modeling," in *Proc. Advance in Neural Inform. Process. System 12*, 2000, pp. 680–686.

[43] M. Wu, "Fighting phishing at the user interface" Ph.D. dissertation, Mass. Inst. Technol., MA, 2004.

[44] R. Xu and D. Wunsch, "Survey of clustering algorithms," *IEEE Trans. Neural Netw.*, vol. 16, no. 3, pp. 645–678, May 2005.

[45] Y. Ye, T. Li, Y. Chen, and Q. Jiang, "Automatic malware categorization using cluster ensemble," in *Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2010, pp. 95–104.

[46] Y. Ye, T. Li, Q. Jiang, Z. Han, and L. Wan, "Intelligent file scoring system for malware detection from the gray list," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2009, pp. 1385–1394.

[47] Y. Ye, D. Wang, T. Li, and D. Ye, "IMDS: Intelligent malware detection system," in *Proc. ACM Int. Conf. Knowl. Discovery Data Mining*, 2007, pp. 1043–1047.

[48] Y. Ye, D. Wang, T. Li, D. Ye, and Q. Jiang, "An intelligent PE-malware detection system based on association mining," *J. Comput. Virol.*, vol. 4, pp. 323–334, Jan. 2008.

[49] Y. Zhang, J. Hong, and L. Cranor, "CANTINA: A content-based approach to detecting phishing web sites," in *Proc. 16th World Wide Web Conf.*, 2007, pp. 639–648.

Authors' photographs and biographies not available at the time of publication.