Drew and Moore *EURASIP Journal on Information Security* (2014) 2014:14
DOI 10.1186/s13635-014-0014-4

EURASIP Journal on Information Security
a SpringerOpen Journal

RESEARCH Open Access

Optimized combined-clustering methods for finding replicated criminal websites

Jake M Drew* and Tyler Moore

Abstract

To be successful, cybercriminals must figure out how to scale their scams. They duplicate content on new websites, often staying one step ahead of defenders that shut down past schemes. For some scams, such as phishing and counterfeit goods shops, the duplicated content remains nearly identical. In others, such as advanced-fee fraud and online Ponzi schemes, the criminal must alter content so that it appears different in order to evade detection by victims and law enforcement. Nevertheless, similarities often remain, in terms of the website structure or content, since making truly unique copies does not scale well. In this paper, we present a novel optimized combined clustering method that links together replicated scam websites, even when the criminal has taken steps to hide connections. We present automated methods to extract key website features, including rendered text, HTML structure, file structure, and screenshots. We describe a process to automatically identify the best combination of such attributes to most accurately cluster similar websites together. To demonstrate the method's applicability to cybercrime, we evaluate its performance against two collected datasets of scam websites: fake escrow services and high-yield investment programs (HYIPs). We show that our method more accurately groups similar websites together than those existing general-purpose consensus clustering methods.

Keywords: Clustering; Consensus clustering; Cybercrime; Escrow fraud; Hierarchical agglomerative clustering; HTML feature extraction; HYIP fraud; Ponzi schemes; High-yield investment programs; Unsupervised learning; Image similarity; Machine learning

1 Introduction

Cybercriminals have adopted two well-known strategies for defrauding consumers online: large-scale and targeted attacks. Many successful scams are designed for massive scale. Phishing scams impersonate banks and online service providers by the thousand, blasting out millions of spam emails to lure a very small fraction of users to fake websites under criminal control [1,2]. Miscreants peddle counterfeit goods and pharmaceuticals, succeeding despite very low conversion rates [3]. The criminals profit because they can easily replicate content across domains, despite efforts to quickly take down content hosted on compromised websites [1]. Defenders have responded by using machine learning techniques to automatically classify malicious websites [4] and to cluster website copies together [5-8].

Given the available countermeasures to untargeted large-scale attacks, some cybercriminals have instead focused on creating individualized attacks suited to their target. Such attacks are much more difficult to detect using automated methods, since the criminal typically crafts bespoke communications. One key advantage of such methods for criminals is that they are much harder to detect until after the attack has already succeeded.

Yet these two approaches represent extremes among available strategies to cybercriminals. In fact, many miscreants operate somewhere in between, carefully replicating the logic of scams without completely copying all material from prior iterations of the attack. For example, criminals engaged in advanced-fee frauds may create bank websites for non-existent banks, complete with online banking services where the victim can log in to inspect their 'deposits'. When one fake bank is shut down, the criminals create a new one that has been tweaked from the former website. Similarly, criminals establish fake escrow services as part of a larger advanced-fee fraud [9]. On the

*Correspondence: jdrew@smu.edu
Computer Science and Engineering Department, Southern Methodist University, 6425 Boaz Lane Dallas, TX 75205, USA

surface, the escrow websites look different, but they often share similarities in page text or HTML structure. Yet another example is online Ponzi schemes called high-yield investment programs (HYIPs) [10]. The programs offer outlandish interest rates to draw investors, which means they inevitably collapse when new deposits dry up. The perpetrators behind the scenes then create new programs that often share similarities with earlier versions.

The designers of these scams have a strong incentive to keep their new copies distinct from the old ones. Prospective victims may be scared away if they realize that an older version of this website has been reported as fraudulent. Hence, the criminals make a more concerted effort to distinguish their new copies from the old ones.

While in principle the criminals could start all over from scratch with each new scam, in practice, it is expensive to recreate entirely new content repeatedly. Hence, things that can be changed easily are (e.g., service name, domain name, registration information). Website structure (if coming from a kit) or the text on a page (if the criminal's English or writing composition skills are weak) are more costly to change, so only minor changes are frequently made.

The purpose of this paper is to design, implement, and evaluate a method for clustering these 'logical copies' of scam websites. Section 2 gives a high-level overview of the combined-clustering process. In Section 3, we describe two sources of data on scam websites used for evaluation: fake escrow websites and HYIPs. Next, Section 4 details how individual website features such as HTML tags, website text, file structure, and image screenshots are extracted to create pairwise distance matrices comparing the similarity between websites. In Section 5, we outline two optimized combined-clustering methods that takes all website features into consideration in order to link disparate websites together. We describe a novel method of combining distance matrices by selecting the minimum pairwise distance. We then evaluate the method compared to other approaches in the consensus clustering literature and cybercrime literature to demonstrate its improved accuracy in Section 6. In Section 7, we apply the method to the entire fake escrow and HYIP datasets and analyze the findings. We review related work in Section 8 and conclude in Section 9.

2 Process for identifying replicated criminal websites

This paper describes a general-purpose method for identifying replicated websites. Figure 1 provides a high-level overview, which is now briefly described before each step is discussed in greater detail in the following sections.

1. *URL crawler:* raw information on websites is gathered.
2. *URL feature extraction:* complementary attributes such as website text and HTML tags are extracted from the raw data for each URL provided.
3. *Input attribute feature files:* extracted features for each website are saved into individual feature files for efficient pairwise distance calculation.
4. *Distance matrices:* pairwise distances between websites for each attribute are computed using the Jaccard distance metrics.
5. *Individual clustering:* hierarchical, agglomerative clustering methods are calculated using each distance matrix, rendering distinct clusterings for each input attribute.
6. *Combined matrices:* combined distance matrices are calculated using various individual distance matrix combinations.
7. *Ground truth selection:* criminal websites are manually divided into replication clusters and used as a source of ground truth.
8. *Cut height optimization:* ground truth clusters are used in combination with the Rand index to identify the optimal clustering cut height for each input attribute.
9. *Combined clustering:* hierarchical, agglomerative clustering methods are calculated using each combined distance matrix to arrive at any number of multi-feature clusterings.
10. *Top performer selection:* the Rand index is calculated for all clusterings against the ground truth to identify the top performing individual feature or combined feature set.

Step 1 is described in Section 3. Steps 2 and 3 are described in Section 4.1, while step 3 is described in Section 4.2. Finally, the clustering steps (5-10) are described in Section 5.

3 Data collection methodology

In order to demonstrate the generality of our clustering approach, we collect datasets on two very different forms of cybercrime: online Ponzi schemes known as HYIPs and fake escrow websites. In both cases, we fetch the HTML using `wget`. We followed links to a depth of 1, while duplicating the website's directory structure. All communications were run through the anonymizing service Tor [11].

3.1 Data source 1: online Ponzi schemes

We use the HYIP websites identified by Moore et al. in [10]. HYIPs peddle dubious financial products that promise unrealistically high returns on customer deposits in the range of 1% to 2% interest, compounded *daily*. HYIPs can afford to pay such generous returns by paying out existing depositors with funds obtained from new

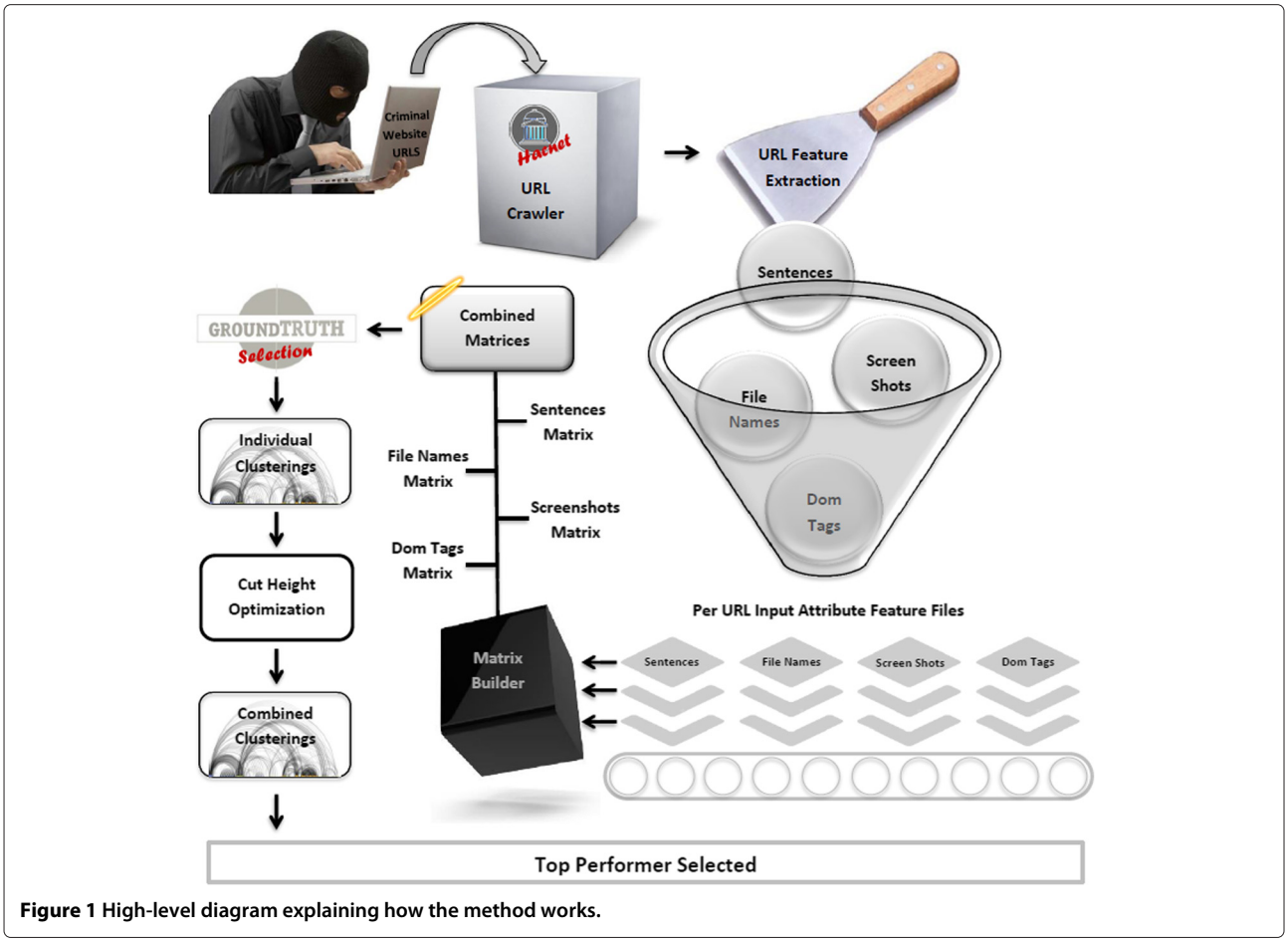

Figure 1 High-level diagram explaining how the method works.

customers. Thus, they meet the classic definition of a Ponzi scheme. Because HYIPs routinely fail, a number of ethically questionable entrepreneurs have spotted an opportunity to track HYIPs and alert investors to when they should withdraw money from schemes prior to collapse. Moore et al. repeatedly crawled the websites listed by these HYIP aggregators, such as `hyip.com`, who monitor for new HYIP websites as well as track those that have failed. In all, we have identified 4,191 HYIP websites operational between 7 November 2010 and 27 September 2012.

3.2 Data source 2: fake escrow websites

A long-running form of advanced-fee fraud is for criminals to set up fraudulent escrow services [9] and dupe consumers with attractively priced high-value items such as cars and boats that cannot be paid for using credit cards. After the sale, the fraudster directs the buyer to use an escrow service chosen by the criminal, which is in fact a sham website. A number of volunteer groups track these websites and attempt to shut the websites down by notifying hosting providers and domain name registrars. We identified reports from two leading sources of fake escrow websites, `aa419.org` and `escrow-fraud.com`. We used automated scripts to check for new reports daily. When new websites are reported, we collect the relevant HTML. In all, we have identified 1,216 fake escrow websites reported between 07 January 2013 and 06 June 2013.

For both data sources, we expect that the criminals behind the schemes are frequently repeat offenders Figure 2. As earlier schemes collapse or are shut down, new websites emerge. However, while there is usually an attempt to hide evidence of any link between the scam websites, it may be possible to identify hidden similarities by inspecting the structure of the HTML code and website content. We next describe a process for identifying such similarities.

4 Identifying and extracting website features

We identified four primary features of websites as potential indicators of similarity: displayed text, HTML tags, directory file names, and image screenshots. These are described in Section 4.1. In Section 4.2, we explain how the features are computed in a pairwise distance matrix.

Figure 2 Examples of replicated website content and file structures for the HYIP dataset.

4.1 Website features

4.1.1 Website text

To identify the text that renders on a given web page, we used a custom 'headless' browser adapted from the WatiN package for C# [12]. We extracted text from all pages associated with a given website, then split the text into sentences using the OpenNLP sentence breaker for C#. Additional lower level text features were also extracted such as character n-grams, word n-grams, and individual words for similarity benchmarking. All text features were placed into individual bags by website. Bags for each website were then compared to create pairwise distance matrices for clustering.

4.1.2 HTML content

Given that cybercriminals frequently rely on kits with similar underlying HTML structure [13], it is important to check the underlying HTML files in addition to the rendered text on the page. A number of choices exist, ranging from comparing the document object model (DOM) tree structure to treating tags on a page as a set of values. From experimentation, we found that DOM trees were too specific, so that even slight variations in otherwise similar pages yielded different trees. We also found that sets of tags did not work well, due to the limited variety of unique HTML tags. We found a middle way by counting how often a tag was observed in the HTML files.

All HTML tags in the website's HTML files were extracted, while noting how many times each tag occurs. We then constructed a compound tag with the tag name and its frequency. For example, if the '
' tag occurs 12 times within the targeted HTML files, the extracted feature value would be '
12'.

4.1.3 File structure

We examined the directory structure and file names for each website since these could betray structural similarity, even when the other content has changed. However, some subtleties must be accounted for during the extraction of this attribute. First, the directory structure is incorporated into the file name (e.g., `admin/home.html`). Second, since most websites include a home or main page given the same name, such as `index.htm`, `index.html`, or `Default.aspx`, websites comprised of only one file may in fact be quite different. Consequently, we exclude the common home page file names from consideration for all websites. Unique file names were placed into bags by website, and pairwise distances were calculated between all websites under consideration.

4.1.4 Website screenshot images

Finally, screenshots were taken for each website using the Selenium automated web browser for C# [14]. Images were resized to $1,000 \times 1,000$ pixels. We calculated both vertical and horizontal luminosity histograms for each

image. Image luminosity features and similarity measures were determined using the EyeOpen image library for C# [15]. During image feature extraction, the red, green, and blue channels for each image pixel were isolated to estimate relative luminance, and these values were then aggregated by each vertical and horizontal image pixel row to calculate two luminosity histograms for each image.

4.2 Constructing distance matrices

For each input attribute, excluding images, we calculated both the Jaccard and Cosine distances between all pairs of websites creating pairwise distance matrices for each input attribute and distance measure. During evaluation, it was determined that the Jaccard distance was the most accurate metric for successfully identifying criminal website replications.

The Jaccard distance between two sets S and T is defined as $1 - J(S, T)$, where

$$J(S, T) = \frac{|S \cap T|}{|S \cup T|}$$

Consider comparing website similarity by sentences. If website A has 50 sentences in the text of its web pages and website B has 40 sentences, and they have 35 sentences in common, then the Jaccard distance is $1 - J(A, B) = 1 - \frac{35}{65} = 0.46$.

Website screenshot images were compared for both vertical and horizontal similarity using luminosity histograms. The luminosity histograms for each matched image pair were compared for similarity by calculating the weighted mean between both the vertical and horizontal histograms. Next, both the average and maximum similarity values between histograms were empirically evaluated for clustering accuracy. Taking the average similarity score between the vertical and horizontal histograms performed best during our evaluation. Once the average vertical and horizontal similarity score was determined, then the pairwise image distance was calculated as 1 - the pairwise image similarity.

Distance matrices were created in parallel for each input attribute by 'mapping' website input attributes into pairwise matches, and then simultaneously 'reducing' pairwise matches into distances using the appropriate distance metric. The pairwise distance matrices were chosen as the output since they are the required input for the hierarchical agglomerative clustering process used during optimized clustering.

5 Optimized combined-clustering process

Once we have individual distance matrices for each input attribute as described in the previous section, the next step is to build the clusters. We first describe two approaches for automatically selecting cut heights for agglomerative clustering: *dynamic cut height*, which is unsupervised, and *optimized cut height*, which is supervised. Next, we compute individual clusterings based on each input attribute. Finally, we construct combined distance matrices for combinations of input attributes and cluster based on the combined matrices.

5.1 Cluster cut height selection

We use a hierarchical agglomerative clustering algorithm [16] to cluster the websites based on the distance matrices. During HAC, a cut height parameter is required to determine the dissimilarity threshold at which clusters are allowed to be merged together. This parameter greatly influences the clustering accuracy, as measured by the Rand index, of the final clusters produced. For instance, using a very high cut height or dissimilarity threshold would result in most websites being included in one giant cluster since a weak measure of similarity is enforced during the merging process.

Traditionally, a static cut height is selected based on the type of data being clustered. Because website input attributes can have very different similarities and still be related, we deploy two methods for automatically selecting the optimal cut heights, one unsupervised and one supervised. In instances where no dependable source of ground truth data is readily available, we use a *dynamic cut height* based on the algorithm used as described in [17]. While the dynamic cut height produces satisfactory results when no ground truth information is available, a better strategy is available where reliable sources of ground truth are present.

Using *optimized cut height*, the best choice is found using the Rand index as a performance measure for each possible cut height parameter value from 0.01 to 0.99. This approach performs clustering and subsequent Rand index scoring at all possible dendrogram height cutoffs using supervised cut height training on the ground truth data. The resulting cut height selected represents the dissimilarity threshold which produces the most accurate clustering results against the ground truth data according to the Rand index score. For example, fake escrow website HTML tags produce clusterings with the Rand index scores ranging from 0% to 97.9% accuracy while varying only the cut height parameter. Figure 3 shows fake escrow website HTML tags generating the highest Rand index score of 0.979 at a cut height of 0.73 with the Rand index score quickly descending back to 0 as the cut height is increased from 0.73 to 1.00. Other fake escrow website input attributes, such as sentences, file names, and images, produce their highest Rand index scores at differing cut height values (0.86, 0.67, and 0.29, respectively).

These results detailed in Section 6.2 demonstrate that the optimized cut height approach produces more accurate clusters than dynamic cut height selection, provided that suitable ground truth data is available to find the

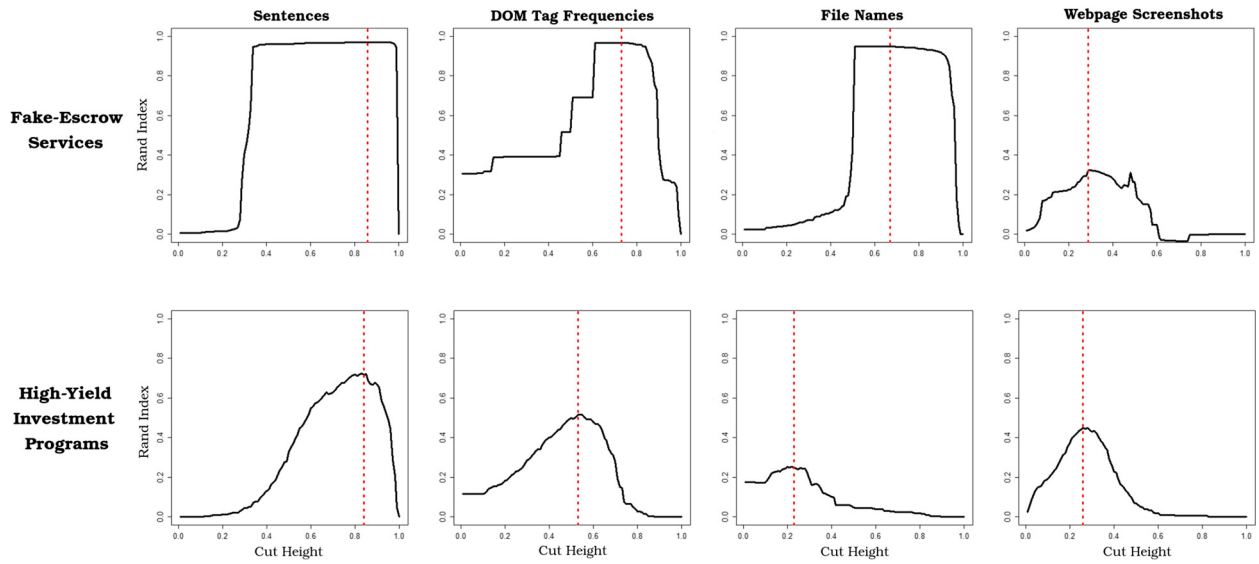

Figure 3 Rand index values for each input attribute at various cut heights.

recommended heights. Furthermore, we also note that the optimized cut height approach performs more consistently, selecting the same top performing input attributes during training and testing executions on both data populations.

5.2 Individual clustering

Because different categories of criminal activity may betray their likenesses in different ways, we need a general process that can select the best combination of input attributes for each dataset. We cannot know, *a priori*, which input attributes are most informative in revealing logical copies. Hence, we start by clustering on each individual attribute independently, before combining the input attributes as described below. It is indeed quite plausible that a single attribute better identifies clusters than does a combination. The clusters are selected using the two cut-height methods outlined above.

5.3 Best min combined clustering

While individual features can often yield highly accurate clustering results, different individual features or even different combinations of multiple features may perform better across different populations of criminal websites as our results will show. Combining multiple distance matrices into a single 'merged' matrix could be useful when different input attributes are important.

However, combining orthogonal distance measures into a single measure must necessarily be an information-lossy operation. A number of other consensus-clustering methods have been proposed [18-21], yet as we will demonstrate in the next section, these algorithms do not perform well when linking together replicated scam websites, often yielding less accurate results than clusterings based on individual input attributes.

Consequently, we have developed a simple and, in practice, more accurate approach to combining the different distance matrices. We define the pairwise distance between two websites a and b as the *minimum* distance across all input attributes. The rationale for doing so is that a website may be very different across one measure but similar according to another. Suppose a criminal manages to change the textual content of many sentences on a website, but uses the same underlying HTML code and file structure. Using the minimum distance ensures that these two websites are viewed as similar. Figure 2 demonstrates examples of both replicated website content and file structures. The highlighted text and file structures for each website displayed are nearly identical. One could also imagine circumstances in which the average or maximum distance among input attributes was more appropriate. We calculate those measures, too but found that the minimum approach worked best and so only those results are reported.

We created combined distance matrices for all possible combinations of distance matrices. In the case of the four input attributes considered in this paper, that means, we produced 11 combined matrices (sentences and DOM tags, sentences and file structures, sentences and images, DOM tags and file structures, DOM tags and images, file structure and images, sentences and DOM tags and file structure, sentences and DOM tags and images, sentences and file structures and images, DOM tags and file structures and images, and sentences and DOM tags and file structures and images). In situations where many

additional features are used, several specifically targeted feature combinations could also be identified for creating a limited number of combined distance matrices.

Combined clusterings are computed for each combined distance matrix using both cut-height selection methods. Ultimately, the top performing individual attribute or combination is selected based on the accuracy observed when evaluating the labeled training dataset.

6 Evaluation against ground truth data

One of the fundamental challenges of clustering logical copies of criminal websites is the lack of ground truth data for evaluating the accuracy of automated methods. Some researchers have relied on expert judgment to assess similarity, but most forego any systematic evaluation due to a lack of ground truth (e.g., [22]). We now describe a method for constructing ground truth datasets for samples of fake escrow services and high-yield investment programs.

We developed a software tool to expedite the evaluation process. This tool enabled pairwise comparison of website screenshots and input attributes (i.e., website text sentences, HTML tag sequences, and file structure) by an evaluator.

6.1 Performing manual ground truth clusterings

After the individual clusterings were calculated for each input attribute, websites could be sorted to identify manual clustering candidates which were placed in the exact same clusters for each individual input attribute's automated clustering. Populations of websites placed into the same clusters for all four input attributes were used as a starting point in the identification of the manual ground truth clusterings. These websites were then analyzed using the comparison tool in order to make a final assessment of whether the website belonged to a cluster. Multiple passes through the website populations were performed in order to place them into the correct manual ground truth clusters. When websites were identified but did not belong in their original assigned cluster, these sites were placed into the unassigned website population for further review and other potential clustering opportunities.

Deciding when to group together similar websites into the same cluster is inherently subjective. We adopted a broad definition of similarity, in which sites were grouped together if they shared most, but not all, of their input attributes in common. Furthermore, the similarity threshold only had to be met for one input attribute. For instance, HYIP websites are typically quite verbose. Many such websites contain three or four identical paragraphs of text, along with perhaps one or two additional paragraphs of completely unique text. For the ground truth evaluation, we deemed such websites to be in the same cluster. Likewise, fake escrow service websites might appear visually identical in basic structure for most of the site. However, a few of the websites assigned to the same cluster might contain extra web pages not present in the others.

We note that while our approach does rely on individual input attribute clusterings as a starting point for evaluation, we do not consider the final combined clustering in the evaluation. This is to maintain a degree of detachment from the combined-clustering method ultimately used on the datasets. We believe the manual clusterings identify a majority of clusters with greater than two members. Although the manual clusterings contain some clusters including only two members, manual clustering efforts were ended when no more clusters of greater than two members were being identified.

6.2 Results

In total, we manually clustered 687 of the 4,188 HYIP websites and 684 of the 1,220 fake escrow websites. The manually clustered websites were sorted by the date each website was identified, and then both datasets were divided into training and testing populations of 80% and 20%, respectively. The test datasets represented 20% of the most recent websites identified within both the fake escrow services and HYIP datasets. Both datasets were divided in this manner to effectively simulate the optimized combined-clustering algorithm's performance in a real-world setting.

In such a scenario, ground truth data would be collected for some period of time and used as training data. Once the training dataset was complete, Rand index optimized cut heights and top performing individual or combined input attributes would be selected using the training data. Going forward, the optimized cut heights would be used during optimized combined-clustering to cluster all new websites identified using the top performing individual or combined input attribute matrices. Chronologically splitting the training and test data in this manner is consistent with how we expect operators fighting cybercrime to use the method.

We computed an adjusted Rand index [23] to evaluate the combined-clustering method described in Section 5 against the constructed ground truth datasets using an optimized cut height which was determined from the training datasets. The optimized cut height was identified by empirically testing cut height values between 0.01 and 0.99 in increments of 0.01 against the training data. Figure 3 illustrates the Rand index values by input attribute at each of these intervals. The optimized Rand index value selected is indicated by the dotted line on each input attribute's chart. Finally, the cut heights selected during the training phase are used to perform optimized combined clustering against the testing data to assess how

this technique might perform in the real-world setting previously described above. We also evaluated employing the unsupervised dynamic tree cut using the method described in [17] to determine an appropriate cut height along with other consensus-clustering methods for comparison. Rand index scores range from 0 to 1, where a score of 1 indicates a perfect match between distinct clusterings.

Table 1 shows the adjusted Rand index for both datasets and all combinations of input attributes using the dynamic and optimized-cut height combined-clustering methods. The first four rows show the Rand index for each individual clustering. For instance, for fake escrow services, clustering based on HTML tags alone using a dynamically determined cut height yielded a Rand index of 0.678 for the training population. Thus, clustering based on tags alone is much more accurate than by website sentences, file structure, or image similarity alone (Rand indices of 0.107, 0.094, and 0.068, respectively). When combining these input attributes, however, we see further

Table 1 Adjusted Rand index for different clusterings, varying the number of input attributes considered (best-performing clusterings italicized)

Scam websites	Dynamic cut height		Optimized cut height	
	Test	Train	Test	Train
Fake escrow services				
Sentences	0.107	0.289	*0.982*	*0.924*
DOM tags	0.678	*0.648*	0.979	0.919
File names	0.094	0.235	0.972	0.869
Images	0.068	0.206	0.325	0.314
S and D	*0.942*	0.584	*0.982*	*0.925*
S and F	0.120	0.245	0.980	0.895
S and I	0.072	0.257	0.962	0.564
D and F	0.558	0.561	0.979	0.892
D and I	0.652	0.614	0.599	0.385
F and I	0.100	0.224	0.518	0.510
S and D and F	0.913	0.561	0.980	0.895
S and D and I	0.883	0.536	0.971	0.673
S and F and I	0.100	0.214	0.975	0.892
D and F and I	0.642	0.536	0.831	0.772
S and D and F and I	*0.941*	0.536	0.971	0.683
High-yield investment programs				
Sentences	*0.713*	*0.650*	*0.738*	*0.867*
DOM tags	0.381	0.399	0.512	0.580
File names	0.261	0.299	0.254	0.337
Images	0.289	0.354	0.434	0.471
S and D	0.393	0.369	0.600	0.671
S and F	0.291	0.310	0.266	0.344
S and I	0.290	0.362	0.437	0.471
D and F	0.309	0.358	0.314	0.326
D and I	0.302	0.340	0.456	0.510
F and I	0.296	0.289	0.397	0.336
S and D and F	0.333	0.362	0.319	0.326
S and D and I	0.319	0.350	0.459	0.510
S and F and I	0.303	0.289	0.398	0.336
D and F and I	0.320	0.337	0.404	0.405
S and D and F and I	0.320	0.337	0.404	0.405

improvement. Clustering based on taking the minimum distance between websites according to HTML tags and sentences yield a Rand index of 0.942, while taking the minimum of all input attributes yields an adjusted Rand index of 0.941. Both combined scores far exceed the Rand indices for any of the other individual input attributes using a dynamically determined cut height.

Results on the test population, for fake escrow services, show that using the dynamic cut height method may not always produce consistent performance results. While the combined matrices achieve the highest Rand index during training, individual HTML tags outperformed all other input attributes by a large margin at 0.648 in the test population.

The optimized cut height algorithm, however, consistently demonstrates a more stable performance selecting the individual sentences matrix and the combined sentences and HTML tags matrix as the top performers in both the training and test populations.

Because cybercriminals act differently when creating logical copies of website for different types of scams, the input attributes that are most similar can change. For example, for HYIPs, we can see that clustering by website sentences yields the most accurate individual Rand index, instead of HTML tags as is the case for fake escrow services. We can also see that for some scams, combining input attributes does not yield a more accurate clustering. Clustering based on the minimum distance of all four attributes yields a Rand index of 0.405 on the optimized cut height's test population, far worse than clustering based on website sentences alone. This underscores the importance of evaluating the individual distance scores against the combined scores, since in some circumstances an individual input attribute or a combination of a subset of the attributes may fare better.

However, it is important to point out that the optimized cut height algorithm appears to more consistently select top performing input matrices and higher Rand index scores on all of the data we benchmarked against. Rand index scores dropped in both the fake escrow services and HYIP test datasets using a dynamically determined cut height (0.294 and 0.63, respectively). When using optimized combined clustering, however, this decrease was smaller in the fake escrow services test population at 0.057 while test results for the HYIP data actually improved from 0.738 to 0.867 for an increase of 0.129% or 12.9%.

We used several general-purpose consensus-clustering methods from R Clue package [24] as benchmarks against the our 'best min optimized-cut height' approach:

1. *'SE'* - implements 'a fixed-point algorithm for obtaining soft least squares Euclidean consensus partitions' by minimizing using Euclidean dissimilarity [19,24].
2. *'DWH'* - uses an extension of the greedy algorithm to implement soft least squares Euclidean consensus partitions [19,24].
3. *'GV3'* - utilizes a sequential unconstrained minimization technique (SUMT) algorithm which is equivalent to finding the membership matrix m for which the sum of the squared differences between $C(m) = mm'$ and the weighted average co-membership matrix $\sum_b w_b C(_m b)$ of the partitions is minimal [20,24].
4. *'soft/symdiff'* - given a maximal number of classes, uses an SUMT approach to minimize using Manhattan dissimilarity of the co-membership matrices coinciding with symdiff partition dissimilarity in the case of hard partitions [21,24].

Table 2 summarizes the best performing measures for the different combined- and consensus-clustering approaches. We can see that our 'best min optimized cut height' approach performs best. It yields more accurate results than other general-purpose consensus-clustering methods, as well as the custom clustering method used to group spam-advertised websites by the authors of [6].

7 Examining the clustered criminal websites

We now apply the dynamic cut-height clustering methods presented earlier to the entire fake escrow (considering sentences, DOM tags, and file structure) and HYIP datasets (considering sentences alone). The 4,191 HYIP websites formed 864 clusters of at least size two, plus an additional 530 singletons. The 1,216 fake escrow websites observed between January and June 2013 formed 161 clusters of at least size two, plus seven singletons.

7.1 Evaluating cluster size

We first study the distribution of cluster size in the two datasets. Figure 4(left panel) plots a CDF of the cluster

Table 2 The best performing measures for the different combined and consensus-clustering approaches (clusterings chosen by the method are italicized)

	Escrow	HYIPs
Minimum	0.683	0.405
Average	0.075	0.443
Max	0.080	0.623
Best min.	*0.985*	*0.867*
DISTATIS	0.070	0.563
Clue SE	0.128	0.245
Clue DWH	0.126	0.472
Clue GV3	0.562	0.508
Clue soft/symdiff	0.095	0.401
Click trajectories [6]	0.022	0.038

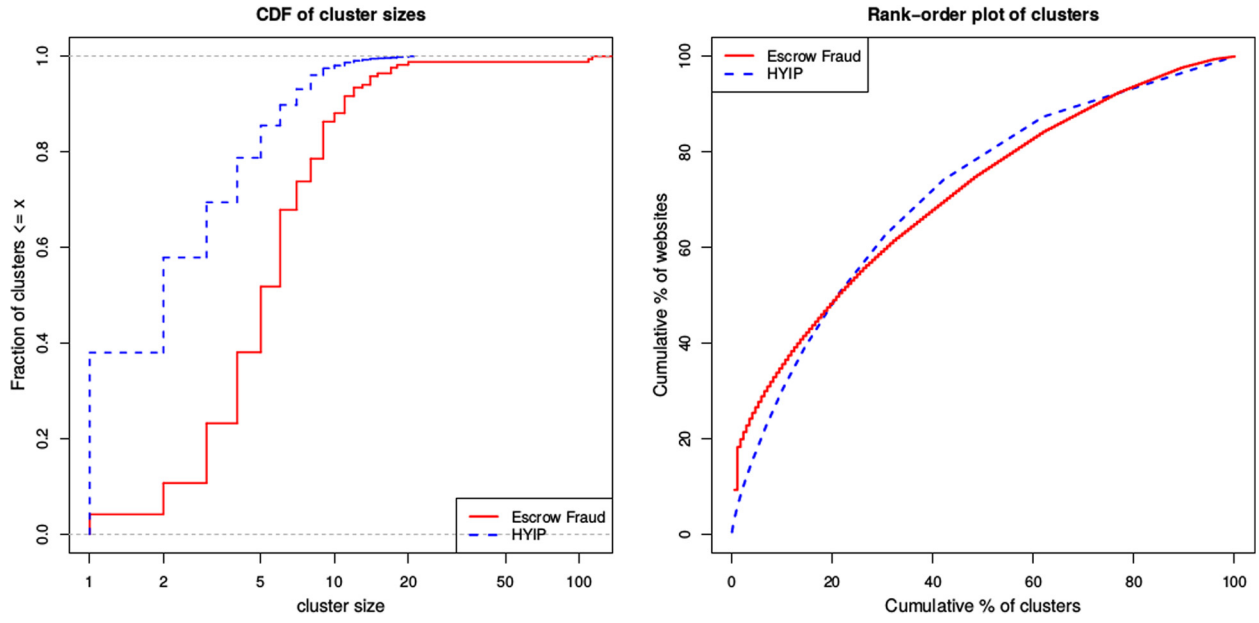

Figure 4 Evaluating the distribution of cluster size in the escrow fraud and HYIP datasets. Cumulative distribution function of cluster size (left panel). Rank order plot of cluster sizes (right panel).

size (note the logarithmic scale on the x-axis). We can see from the blue dashed line that the HYIPs tend to have smaller clusters. In addition to the 530 singletons (40% of the total clusters), 662 clusters (47% of the total) include between two and five websites. One hundred seventy-five clusters (13%) are sized between six and ten websites, with 27 clusters including more than ten websites. The biggest cluster included 20 HYIP websites. These results indicate that duplication in HYIPs, while frequent, does not occur on the same scale as many other forms of cybercrime.

There is more overt copying in the fake escrow dataset. Only seven of the 1,216 escrow websites could not be clustered with another website. Eighty clusters (28% of the total) include between two and five websites, but another 79 clusters are sized between six and 20. Furthermore, two large clusters (including 113 and 109 websites, respectively) can be found. We conclude that duplication is used more often as a criminal tactic in the fake escrow websites than for the HYIPs.

Another way to look at the distribution of cluster sizes is to examine the rank-order plot in Figure 4(right panel). Again, we can observe differences in the structure of the two datasets. Rank-order plots sort the clusters by size and show the percentages of websites that are covered by the smallest number of clusters. For instance, we can see from the red solid line the effect of the two large clusters in the fake escrow dataset. These two clusters account for nearly 20% of the total fake escrow websites. After that, the next biggest clusters make a much smaller contribution in identifying more websites. Nonetheless, the incremental contributions of the HYIP clusters (shown in the dashed blue line) are also quite small. This relative dispersion of clusters differs from the concentration found in other cybercrime datasets where there is large-scale replication of content.

7.2 Evaluating cluster persistence

We now study how frequently the replicated criminal websites are re-used over time. One strategy available to criminals is to create multiple copies of the website in parallel, thereby reaching more victims more quickly. The alternative is to re-use copies in a serial fashion, introducing new copies only after time has passed or the prior instances have collapsed. We investigate both datasets to empirically answer the question of which strategy is preferred.

Figure 5 groups the ten largest clusters from the fake escrow dataset and plots the date at which each website in the cluster first appears. We can see that for the two largest clusters there are spikes where multiple website copies are spawned on the same day. For the smaller clusters, however, we see that websites are introduced sequentially. Moreover, for all of the biggest clusters, new copies are introduced throughout the observation period. From this, we can conclude that criminals are likely to use the same template repeatedly until stopped.

Next, we examine the observed persistence of the clusters. We define the 'lifetime' of a cluster as the difference in days between the first and the last appearance of a website

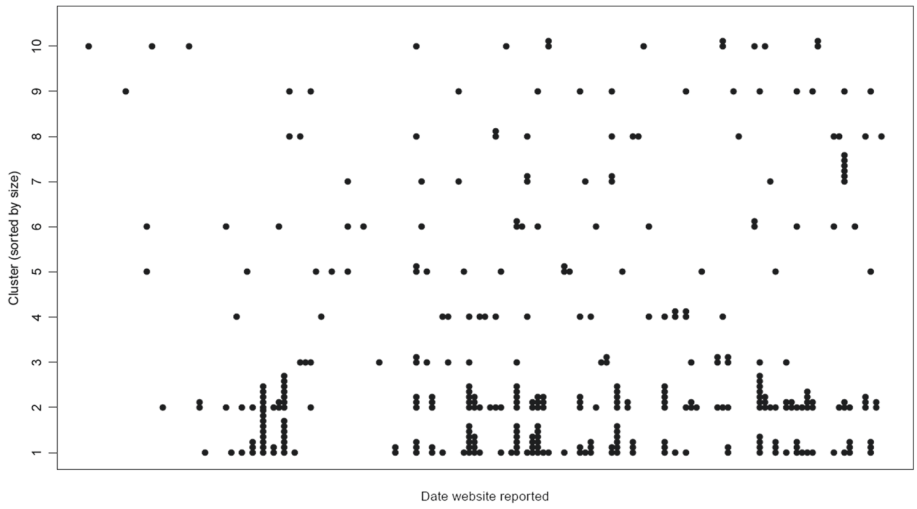

Figure 5 Top ten largest clusters in the fake escrow dataset by date the websites are identified.

in the cluster. For instance, the first reported website in one cluster of 18 fake escrow websites appeared on 2 February 2013, while the last one occurred on 7 May 2013. Hence, the lifetime of the cluster is 92 days. Longer-lived clusters indicate that cybercriminals can create website copies for long periods of time with impunity.

We use a survival probability plot to examine the distribution of cluster lifetimes. A survival function $S(t)$ measures the probability that a cluster's lifetime is greater than time t. Survival analysis takes into account 'censored' data points, i.e., when the final website in the cluster is reported near the end of the study. We deem any cluster with a website reported within 14 days of the end of data collection to be censored. We use the Kaplan-Meier estimator [25] to calculate a survival function.

Figure 6 gives the survival plots for both datasets (solid lines indicate the survival probability, while dashed lines indicate 95% confidence intervals). In the left graph, we can see that around 75% of fake escrow clusters persist for at least 60 days, and that the median lifetime is 90 days. Note that around 25% of the clusters remained active at the end of the 150-day measurement period, so we cannot reason about how long these most persistent clusters will remain.

Because we tracked HYIPs for a much longer period (Figure 6 (right)), nearly all clusters eventually ceased to be replicated. Consequently, the survival probability for even long-lived clusters can be evaluated. Twenty percent of the HYIP clusters persist for more than 500 days, while 25% do not last longer than 100 days. The median lifetime of HYIP clusters is around 250 days. The relatively long persistence of many HYIP clusters should give law enforcement some encouragement: because the criminals reuse content over long periods, tracking them down becomes a more realistic proposition.

8 Related work

A number of researchers have applied machine learning methods to cluster websites created by cybercriminals. Wardman et al. examined the file structure and content of suspected phishing web pages to automatically classify reported URLs as phishing [7]. Layton et al. cluster phishing web pages together using a combination of k-means and agglomerative clustering [8].

Several researchers have classified and clustered web spam pages. Urvoy et al. use HTML structure to classify web pages, and they develop a clustering method using locality-sensitive hashing to cluster similar spam pages together [26]. Lin uses HTML tag multisets to classify cloaked web pages [27]. Lin's technique is used by Wang et al. [28] to detect when the cached HTML is very different from what is presented to the user. Finally, Anderson et al. use image shingling to cluster screenshots of websites advertised in email spam [5]. Similarly, Levchenko et al. use a custom clustering heuristic method to group similar spam-advertised web pages [6]. We implemented and evaluated this clustering method on the cybercrime datasets in Section 6. Der et al. clustered storefronts selling counterfeit goods by the affiliate structure driving traffic to different stores [29]. Finally, Leontiadis et al. group similar unlicensed online pharmacy inventories [22]. They did not attempt to evaluate against ground truth; instead they used the Jaccard distance and agglomerative clustering to find suitable clusters.

Neisius and Clayton also studied high-yield investment programs [13]. Notably, they estimated that a majority

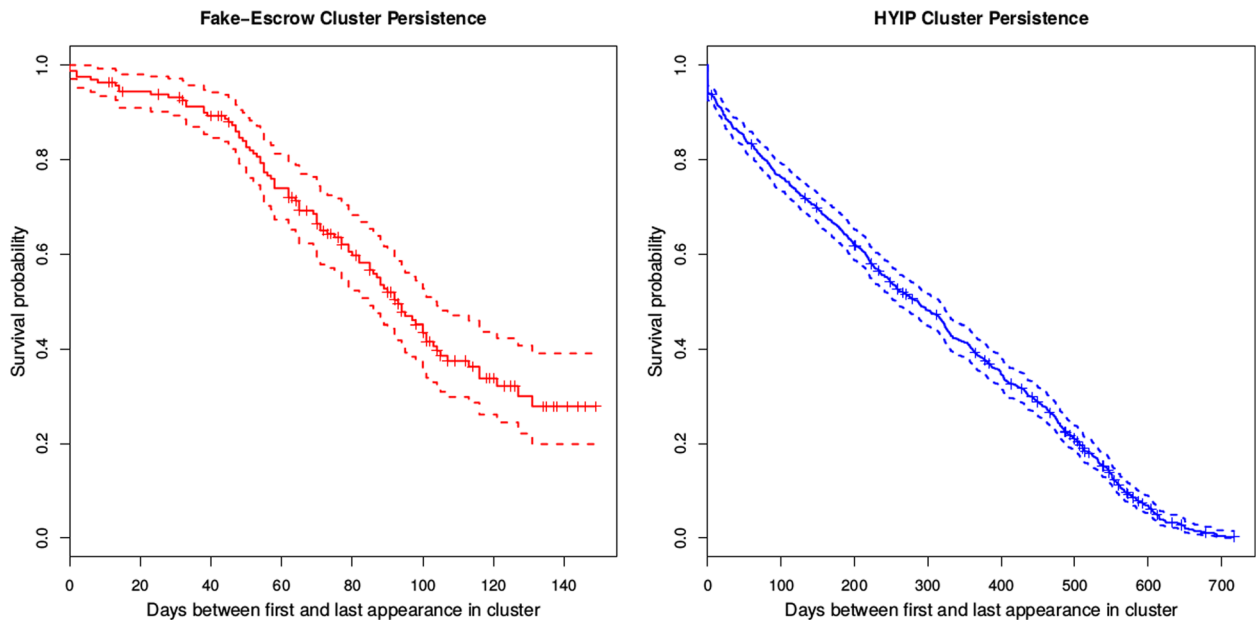

Figure 6 Survival probability of fake escrow clusters (left) and HYIP clusters (right).

of HYIP websites used templates licensed from a company called 'Goldcoders'. While we did observe some Goldcoder templates in our own datasets, we did not find them occurring at the same frequency. Furthermore, our clustering method tended to link HYIP websites more by the rendered text on the page rather than the website file structure.

Separate to the work on cybercriminal datasets, other researchers have proposed consensus clustering methods for different applications. DISTATIS is an adaptation of the STATIS methodology specifically used for the purpose of integrating distance matrices for different input attributes [30]. DISTATIS can be considered a three-way extension of metric multidimensional scaling [31], which transforms a collection of distance matrices into cross-product matrices used in the cross-product approach to STATIS. Consensus can be performed between two or more distance matrices by using DISTATIS and then converting the cross-product matrix output into into a (squared) Euclidean distance matrix which is the inverse transformation of metric multidimensional scaling [32].

Our work follows in the line of both of the above research thrusts. It differs in that it considers multiple attributes that an attacker may change (site content, HTML structure, and file structure), even when she may not modify all attributes. It is also tolerant of greater changes by the cybercriminal than previous approaches. At the same time, though, it is more specific than general consensus clustering methods, which enables the method to achieve higher accuracy in cluster labelings.

9 Conclusions

When designing scams, cybercriminals face trade-offs between scale and victim susceptibility and between scale and evasiveness from law enforcement. Large-scale scams cast a wider net, but this comes at the expense of lower victim yield and faster defender response. Highly targeted attacks are much more likely to work, but they are more expensive to craft. Some frauds lie in the middle, where the criminals replicate scams but not without taking care to give the appearance that each attack is distinct.

In this paper, we propose and evaluate a combined-clustering method to automatically link together such semi-automated scams. We have shown it to be more accurate than general-purpose consensus-clustering approaches, as well as approaches designed for large-scale scams such as phishing that use more extensive copying of content. In particular, we applied the method to two classes of scams: HYIPs and fake escrow websites.

The method could prove valuable to law enforcement, as it helps tackle cybercrimes that individually are too minor to investigate but collectively may cross a threshold of significance. For instance, our method identifies two distinct clusters of more than 100 fake escrow websites each. Furthermore, our method could substantially reduce the workload for investigators as they prioritize which criminals to investigate.

Competing interests
The authors declare that they have no competing interests.

Acknowledgements
We would like to thank the operators of `escrow-fraud.com` and `aa419.org` for allowing us to use their lists of fake escrow websites.
This work was partially funded by the Department of Homeland Security (DHS) Science and Technology Directorate, Cyber Security Division (DHS S&T/CSD) Broad Agency Announcement 11.02, the Government of Australia, and the SPAWAR Systems Center Pacific via contract number N66001-13-C-0131. This paper represents the position of the authors and not that of the aforementioned agencies.

Received: 10 August 2014 Accepted: 12 September 2014
Published online: 16 December 2014

References

1. T Moore, R Clayton, in *Second APWG eCrime Researchers Summit. eCrime '07*. Examining the impact of website take-down on phishing (ACM Pittsburgh, 2007)
2. D Florencio, C Herley, in *Second APWG eCrime Researchers Summit. eCrime '07*. Evaluating a trial deployment of password re-use for phishing prevention (ACM New York, 2007), pp. 26–36. doi:10.1145/1299015.1299018. http://doi.acm.org/10.1145/1299015.1299018
3. C Kanich, C Kreibich, K Levchenko, B Enright, G Voelker, V Paxson, S Savage, in *Conference on Computer and Communications Security (CCS)*. Spamalytics: an empirical analysis of spam marketing conversion (Alexandria, VA, 2008)
4. N Provos, P Mavrommatis, M Rajab, F Monrose, in *17th USENIX Security Symposium*. All your iFrames point to us, (2008)
5. DS Anderson, C Fleizach, S Savage, GM Voelker, in *Proceedings of 16th USENIX Security Symposium*. Spamscatter: Characterizing Internet scam hosting infrastructure (USENIX Association Berkeley, 2007), pp. 10–11014. http://dl.acm.org/citation.cfm?id=1362903.1362913
6. K Levchenko, A Pitsillidis, N Chachra, B Enright, M Félegyházi, C Grier, T Halvorson, C Kanich, C Kreibich, H Liu, D McCoy, N Weaver, V Paxson, GM Voelker, S Savage, in *Proceedings of the 2011 IEEE Symposium on Security and Privacy. SP '11*. Click trajectories: end-to-end analysis of the spam value chain (IEEE Computer Society Washington, DC, 2011), pp. 431–446. doi:10.1109/SP.2011.24. http://dx.doi.org/10.1109/SP.2011.24
7. B Wardman, G Warner, in *eCrime Researchers Summit, 2008*. Automating phishing website identification through deep MD5 matching (IEEE, 2008), pp. 1–7
8. R Layton, P Watters, R Dazeley, in *eCrime Researchers Summit (eCrime), 2010*. Automatically determining phishing campaigns using the uscap methodology, (2010), pp. 1–8. doi:10.1109/ecrime.2010.5706698
9. T Moore, R Clayton, *The Impact of Incentives on Notice and Take-down*. (ME Johnson, ed.) (Springer, 2008), pp. 199–223
10. T Moore, J Han, R Clayton, in *Financial Cryptography*. Lecture Notes in Computer Science, vol. 7397, ed. by Keromytis A D. The postmodern Ponzi scheme: Empirical analysis of high-yield investment programs (Springer, 2012), pp. 41–56. http://lyle.smu.edu/~tylerm/fc12.pdf
11. R Dingledine, N Mathewson, P Syverson, in *13th USENIX Security Symposium*. Tor: The second-generation onion router, (2004)
12. WatiN: Web application Testing in.Net. http://www.watin.org Accessed October 16, 2014
13. J Neisius, R Clayton, in *APWG Symposium on Electronic Crime Research*. Orchestrated crime: the high yield investment fraud ecosystem, (2014)
14. HQ Selenium. http://www.seleniumhq.org/ Accessed October 16, 2014
15. Similar images finder - ṄET Image processing in C# and RGB projections. https://similarimagesfinder.codeplex.com/ Accessed October 16, 2014
16. SC Johnson, Hierarchical clustering schemes. Psychometrika. **32**(3), 241–254 (1967)
17. P Langfelder, B Zhang, S Horvath, Defining clusters from a hierarchical cluster tree. Bioinformatics. **24**(5), 719–720 (2008). doi:10.1093/bioinformatics/btm563
18. H Abdi, AJ O'Toole, D Valentin, B Edelman, in *Computer Vision and Pattern Recognition - Workshops, 2005. CVPR Workshops. IEEE Computer Society Conference On*. Distatis: The analysis of multiple distance matrices, (2005), pp. 42–42. doi:10.1109/CVPR.2005.445
19. E Dimitriadou, A Weingessel, K Hornik, A combination scheme for fuzzy clustering. Int. J. Pattern Recogn. Artif. Intell. **16**(07), 901–912 (2002). doi:10.1142/S0218001402002052. http://www.worldscientific.com/doi/pdf/10.1142/S0218001402002052
20. AD Gordon, M Vichi, Fuzzy partition models for fitting a set of partitions. Psychometrika. **66**(2), 229–247 (2001). doi:10.1007/BF02294837
21. AV Fiacco, GP McCormick, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, vol. 4, (Siam, 1990)
22. N Leontiadis, T Moore, N Christin, in *Proceedings of the Fourteenth ACM Conference on Electronic Commerce*. Pick your poison: pricing and inventories at unlicensed online pharmacies (ACM, 2013), pp. 621–638
23. WM Rand, Objective criteria for the evaluation of clustering methods. J. Am. Stat. Assoc. **66**(336), 846–850 (1971). doi:10.1080/01.621459.1971.10482356
24. K Hornik, A CLUE for CLUster ensembles. Journal of Statistical Software. **14**, 65–72 (2005)
25. EL Kaplan, P Meier, Nonparametric estimation from incomplete observations. J. Am. Stat. Assoc. **53**, 457–481 (1958)
26. T Urvoy, E Chauveau, P Filoche, T Lavergne, Tracking web spam with html style similarities. ACM Trans. Web. **2**(1), 3–1328 (2008). doi:10.1145/1326561.1326564
27. J-L Lin, Detection of cloaked web spam by using tag-based methods. Expert Syst. Appl. **36**(4), 7493–7499 (2009). doi:10.1016/j.eswa.2008.09.056. Available at, http://dx.doi.org/10.1016/j.eswa.2008.09.056
28. DY Wang, S Savage, GM Voelker, in *Proceedings of the 18th ACM Conference on Computer and Communications Security. CCS '11*. Cloak and dagger: dynamics of web search cloaking (ACM New York, 2011), pp. 477–490. doi:10.1145/2046707.2046763. Available at http://doi.acm.org/10.1145/2046707.2046763
29. MF Der, LK Saul, S Savage, Voelker G M, in *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. Knock it off: profiling the online storefronts of counterfeit merchandise (ACM, 2014)
30. H Abdi, LJ Williams, D Valentin, M Bennani-Dosse, Statis and distatis: optimum multitable principal component analysis and three way metric multidimensional scaling. Wiley Interdiscip. Rev.: Comput. Stat. **4**(2), 124–167 (2012). doi:10.1002/wics.198
31. S Krolak-Schwerdt, in *Data Analysis and Decision Support*. Studies in Classification, Data Analysis, and Knowledge Organization, ed. by D Baier, R Decker, and L Schmidt-Thieme. Three-way multidimensional scaling: formal properties and relationships between scaling methods (Springer, 2005), p. 82–90. doi:10.1007/3-540-28397-8_10. http://dx.doi.org/10.1007/3-540-28397-8_10
32. H Abdi, ed. by NJ Salkind. Encyclopedia of Measurement and Statistics (SAGE Publications, Inc, 2007), p. 598–605. doi:10.4135/9781412952644. http://dx.doi.org/10.4135/9781412952644