Detecting Phishing Web Pages with Visual Similarity Assessment Based on Earth Mover's Distance (EMD)

Anthony Y. Fu, Liu Wenyin, *Senior Member*, *IEEE*, and Xiaotie Deng, *Senior Member*, *IEEE*

Abstract—An effective approach to phishing Web page detection is proposed, which uses Earth Mover's Distance (EMD) to measure Web page visual similarity. We first convert the involved Web pages into low resolution images and then use color and coordinate features to represent the image signatures. We use EMD to calculate the signature distances of the images of the Web pages. We train an EMD threshold vector for classifying a Web page as a phishing or a normal one. Large-scale experiments with 10,281 suspected Web pages are carried out to show high classification precision, phishing recall, and applicable time performance for online enterprise solution. We also compare our method with two others to manifest its advantage. We also built up a real system which is already used online and it has caught many real phishing cases.

Index Terms—Antiphishing, visual assessment, Earth Mover's Distance.

1 Introduction

Phishing Web pages are forged Web pages that are created by malicious people to mimic Web pages of real Web sites. Most of these kinds of Web pages have high visual similarities to scam their victims. Some of these kinds of Web pages look exactly like the real ones. Unwary Internet users may be easily deceived by this kind of scam. Victims of phishing Web pages may expose their bank account, password, credit card number, or other important information to the phishing Web page owners.

Phishing is a relatively new Internet crime in comparison with other forms, e.g., virus and hacking. More and more phishing Web pages have been found in recent years in an accelerative way. A report from the Anti-Phishing Working Group [2] shows that the number of phishing Web pages is increasing each month by 50 percent and usually 5 percent of the phishing e-mail receivers will respond to the scams. Also, there were 15,050 phishing cases reported simply in one month in June 2005 [2]. This problem has drawn high attention from both industry and the academic research domain since it is a severe security and privacy problem and has caused huge negative impacts on the Internet world. It is threatening people's confidence to use the Web to conduct online finance-related activities. As phishing e-mail is the major method used to scam Internet users, it will be an obligation for e-mail service providers to protect their users from it. Moreover, governments, law executive bureaus, and the owners of victim Web sites should also be responsible for protecting Internet users from suffering this kind of attack.

In this paper, we propose an effective approach for detecting phishing Web pages, which employs the Earth Mover's Distance (EMD) [13] to calculate the visual similarity of Web pages. The most important reason that Internet users could become phishing victims is that phishing Web pages always have high visual similarity with the real Web pages, such as visually similar block layouts, dominant colors, images, and fonts, etc. We follow the antiphishing strategy in [19] to obtain suspected Web pages, which are supposed to be collected from URLs in those e-mails containing keywords associated with protected Web pages. We first convert them into normalized images and then represent their image signatures with features composed of dominant color category and its corresponding centroid coordinate to calculate the visual similarity of two Web pages. The linear programming algorithm [12] for EMD is applied to visual similarity computation of the two signatures. An antiphishing system may be requested to protect many Web pages. A threshold is calculated for each protected Web page using supervised training. If the EMD-based visual similarity of a Web page exceeds the threshold of a protected Web page, we classify the Web page as a phishing one.

Large-scale experiments have been carried out. Our final experiments show high classification precision, high phishing recall, and satisfactory time performance. We also did a performance comparison with the method proposed by Liu et al. in [17], [18], [19] and another EMD-based method (the HTML-based EMD method) and the results of the experiments show that the proposed method is superior to the other two.

The rest of this paper is organized as follows: In Section 2, related works of other antiphishing methods and relative techniques are introduced. In Section 3, Web page preprocessing and Web page signature is addressed.

• *The authors are with the Department of Computer Science, City University of Hong Kong, 83, Tat Chee Ave., Hong Kong SAR. E-mail: anthony@cs.cityu.edu.hk, and {csliuwy, csdeng}@cityu.edu.hk.*

Manuscript received 12 Oct. 2005; revised 4 May 2006; accepted 7 June 2006; published online 2 Nov. 2006.
For information on obtaining reprints of this article, please send e-mail to: tdsc@computer.org, and reference IEEECS Log Number TDSC-0138-1005.

In Section 4, we present the EMD-based approach to visual similarity measurement. In Section 5, we discuss our phishing classification method. In Section 6, we discuss the experiments of our approach. In Section 7, our visual similarity-based antiphishing prototype system is introduced. Finally, we conclude our work and discuss future works in Section 8.

2 Related Works

Evolving with the antiphishing techniques, various phishing techniques and more complicated and hard-to-detect methods are used by phishers. The most straightforward way for a phisher to scam people is to make the phishing Web pages similar to their targets.

A phishing strategy includes both Web link obfuscation and Web page obfuscation. Web link obfuscation can be carried out in four basic ways:

1. adding a suffix to a domain name of the URL,
2. using an actual link different from the visible link,
3. utilizing system bugs in real Web sites to redirect the link to the phishing Web pages, and
4. using cousin domain names (e.g., replacing certain characters in the target URL with similar characters) [8].

The Web page obfuscation can be carried out in three basic ways:

1. using the downloaded Web page from the real Web site to make the phishing Web page appear and react exactly the same as the real one does,
2. using a script or images to cover the address bar to scam users into believing they have entered the correct Web sites, and
3. using visual-based content (Image, Flash, JavaApplet, etc.) rather than HTML to avoid HTML-based phishing detection.

Intuitively, phishing (or similar) Web page detection is similar to duplicated or plagiarized document detection in some extent and the document similarity evaluation techniques can be used for the antiphishing task. Previous research works on duplicated document detection approaches focus on plain text documents and use pure text features in similarity measure, such as collection statistics [5], syntactic analysis [3], displaying structure [4], [20], [27], visual-based understanding [11], vector space model [24], etc. Hoad and Zobel have surveyed various methods on plagiarized document detection in [14]. However, as Liu et al. [19] demonstrated, pure text features are not sufficient for phishing Web page detection since phishing Web pages mainly employ visual similarity to scam users.

There are many antiphishing techniques popularly used by the industry. The most popular and major antiphishing methods include authentication, which includes e-mail authentication and Web page authentication (e.g., [21]), filtering, which includes e-mail filtering and Web page filtering, attack analyzing and tracing, immune-system-like phishing report and detection, and network law enforcement. Many user interface based antiphishing approaches, including Web browser toolbars, e-mail client agent toolbars, and distributed server applications, are also commonly used. APWG provides a solution directory at [2] which contains most of the major antiphishing companies in the world. However, an automatic antiphishing method is seldom reported.

The typical technologies of antiphishing from the User Interface aspect are done by Dhamija and Tygar in [7] and Wu et al. in [26]. They proposed methods that need Web page creators to follow certain rules to create Web pages, either by adding dynamic skin to Web pages or adding sensitive information location attributes to HTML code. However, it is difficult to convince all Web page creators to follow the rules.

In [17], [18], [19], the DOM-based [25] visual similarity of Web pages is oriented, and the concept of visual approach to phishing detection was first introduced. Through this approach, a phishing Web page can be detected and reported in an automatic way rather than involving too many human efforts. Their method first decomposes the Web pages (in HTML) into salient (visually distinguishable) block regions. The visual similarity between two Web pages is then evaluated in three metrics: block level similarity, layout similarity, and overall style similarity, which are based on the matching of the salient block regions. Our approach in this paper follows the overall strategy in [17], [18], [19] but uses a different way to calculate the visual similarity of Web pages. We first convert HTML Web pages into images and then employ the EMD method to the signatures of the images for similarity calculation. We always have a concern of "what if phishers use images to represent the Web page content rather than HTML text?" Flash, Movie, ActiveX, Java Applet, and various types of pictures can be embedded into the Web pages instead of HTML text. In other words, phishers can easily create Web pages that look exactly the same as the real Web page but use completely different background coding (Text, Flash, ActiveX, Java Applet, and Various Pictures). A real Web page can correspond to countless fake Web pages with different code. That is also our major motivation for investigating the phishing detection method at the pixel level.

EMD [13] is a method to evaluate the distance (dissimilarity) between two signatures. A signature is a set of features and their corresponding weights. The method comes from the well-known transportation problem. Suppose we have m producers and each producer comes with a weight representing the amount of product he has. We denote producer set P as:

$$P = \{(p_1, w_{p_1}), (p_2, w_{p_2}), \ldots, (p_m, w_{p_m})\}. \quad (1)$$

Suppose we also have n customers and each consumer comes with a weight indicating the amount of product he needs. We denote the consumer set C as:

$$C = \{(c_1, w_{c_1}), (c_2, w_{c_2}), \ldots, (c_n, w_{c_n})\}. \quad (2)$$

Producers want to transport their products to consumers. Suppose the distances of each pair of producer and consumer are given, and they are represented into a distance matrix D, which is defined before calculating EMD. It is represented as:

$$D = [d_{ij}], \text{ where } 1 \leq i \leq m \text{ and } 1 \leq j \leq n. \quad (3)$$

Producers produce the same product and consumers consume the same product. The transportation fee is proportional to both distance and product weight. The task is to find a flow matrix F, which contains factors indicating the amount of product to be moved from one producer to one consumer.

$$F = [f_{ij}], \text{ where } 1 \le i \le m \text{ and } 1 \le j \le n. \tag{4}$$

The transported product amount from P to C should be as much as possible and the total transportation fee should be minimized. The total cost of transportation fee can be represented as:

$$\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij} \cdot d_{ij}. \tag{5}$$

The calculation of F is subject to the following constraints:

$$s.t. \begin{cases} f_{ij} \ge 0 & where\ 1 \le i \le m, 1 \le j \le n \\ \sum_{j=1}^{n} f_{ij} \le w_{pi} & where\ 1 \le i \le m \\ \sum_{i=1}^{m} f_{ij} \le w_{cj} & where\ 1 \le j \le n \\ \sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij} = Min\left(\sum_{i=1}^{m} w_{pi}, \sum_{j=1}^{n} w_{cj}\right). \end{cases} \tag{6}$$

It is a Linear Programming problem. We solve it to get F, and then calculate EMD. The EMD can be represented as:

$$EMD(P, C, D) = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n}(f_{ij} \cdot d_{ij})}{\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}}. \tag{7}$$

It has been practically proved that EMD has advantages in representing problems involving multifeatured signatures. EMD allows for partial matches in a very natural way and is especially fit for cognitive distance evaluation, as shown in [16]. People have successfully used it for vision problems [10], [22], [23].

3 Web Page Preprocessing and Signature Generation

We retrieve the suspected Web pages and protected Web pages from the Web and generate their signatures. The task of our Web page preprocessing approach contains three procedures: 1) obtain the image of a Web page from its URL, 2) perform normalization, and 3) represent the Web page image into a Web page visual signature (consists of color and coordinate features), which is used to evaluate the visual similarity of a pair of Web pages.

The process of displaying a Web page in a Web browser on the screen from HTML and accessory files (including pictures, Flash movies, ActiveX plugins, Java Applets, etc.) is the Web page rendering process. We use GDI (graphic device interface) API provided by the Microsoft IE browser to get Web page images (in jpeg format). The images of the original sizes are processed into images with normalized size (e.g., $100 * 100$). The Lanczos algorithm [15] is used to calculate the resized image because the Lanczos algorithm has very strong antialiasing properties in Fourier domain, and it is also easy to be computed in spatial domain. Moreover, sharp images can be generated with the Lanczos algorithm as intuitively, the sharp images could provide better signature for identification from the others. Fig. 1 shows examples of original Web pages and resized images (to $100 * 100$ and $10 * 10$, respectively). www.bbb.org is an example of a square-like image, www.banktechnews.com is an example of a longer image, and www.bankofcyprus.com is an example of a wider image. All of them are normalized into a fixed-size square image, respectively. We use the normalized images to present the signature of each Web page. As there is no method existing to determine what size of the normalized image is the best for representing the original one, we empirically choose a suitable size based on experiments in Section 6.

A signature of an image is a feature vector which can effectively represent the image. The signature of an image in our approach is comprised of features and their corresponding weights. A feature is comprised of a color and the centroid of its position distribution in the image. The color of each pixel in the resized images is represented using the ARGB (alpha, red, green, and blue) scheme with 4 bytes (32 bits). A color can be represented with a 4-tuple $< A, R, G, B >$. However, this is a huge color space, which includes $2^{32} = 4,294,967,296$ colors. In practice, we use a degraded color space to represent the signature of an image. We define the Color Degrading Factor (CDF) to be the scale of each color component making a change. Thus, we have $(2^8/\text{CDF})^4$ colors in our degraded color space. A degraded color can be represented as

$$< A\text{-}(A\ mod\ CDF), B\text{-}(B\ mod\ CDF), C\text{-}(C\ mod\ CDF), D\text{-}(D\ mod\ CDF) > .$$

For example, when $\text{CDF} = 32$, we have 4,096 colors in the degraded color space. The centroid of each degraded color is calculated using $C_{dc} = \sum_{i=1}^{N_{dc}} \frac{c_{dc,i}}{N_{dc}}$, C_{dc} is the centroid of degraded color dc, $c_{dc,i}$ is the coordinates of the $i\text{th}$ pixel that has degraded color dc, and N_{dc} is the total number of pixels that have degraded color dc, i.e., the frequency of dc. A feature F_{dc}, which has degraded color dc, can be represented with dc and C_{dc}, $F_{dc} =< dc, C_{dc} >$. The weight corresponding to this feature is the color's frequency N_{dc}. A complete signature S is represented as

$$S =<< F_{dc_1}, N_{dc_1} >, < F_{dc_2}, N_{dc_2} >, \ldots, < F_{dc_N}, N_{dc_N} >>,$$

where N is the total number of degraded colors. The feature-weight tuples in S are ranked in the descending order of their weights, i.e., $N_{dc_i} \ge N_{dc_{i+1}}$ for $1 \le i \le N-1$. In our approach, we do not use all of the features. We choose the first N_s most frequent colors in S to be the signature, where N_s is less or equal to N, and we denote it as S_s. When N is less than N_s, S is chosen to be exactly S_s.

4 Computing Visual Similarity from EMD

We use EMD to calculate the similarity of two Web pages based on their signatures as follows: The distance matrix $D = [d_{ij}]$ $(1 \le i \le m, 1 \le j \le n)$ is defined in advance using

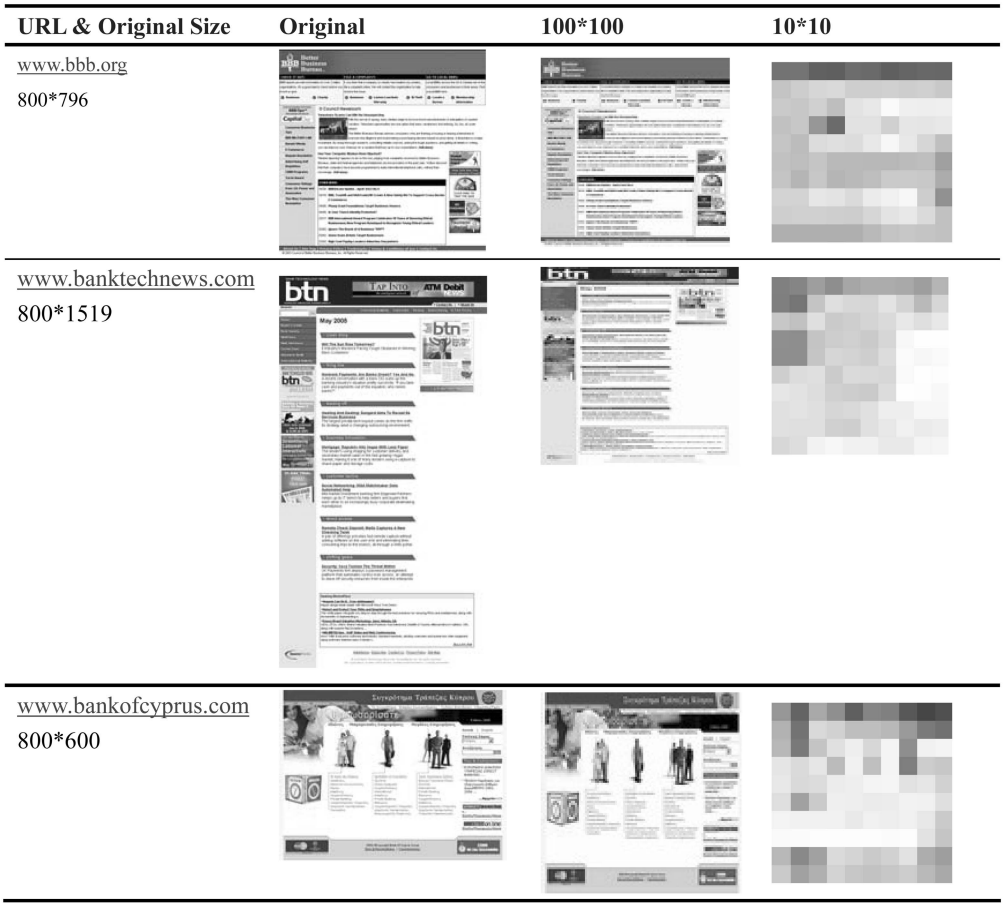


Fig. 1. Examples of Web page preprocessing results.

a straightforward way. We first calculate the normalized euclidian distance of the degraded ARGB colors, and then calculate the normalized Euclidian distance of centroids. The two distances are added up with weights p and q, respectively, to form the feature distance, where $p+q=1$.

Suppose we have feature $\varphi_i =< dc_i, C_{dc_i} >$, where $dc_i =< dA_i, dR_i, dG_i, dB_i >$, feature $\varphi_j =< dc_j, C_{dc_j} >$, where $dc_j =< dA_j, dR_j, dG_j, dB_j >$, the maximum color distance

$$MD_{color} =\| < MaxA-0, MaxR-0, \\ MaxG-0, MaxB-0 > \|,$$

where $MaxA$, $MaxR$, $MaxG$, and $MaxB$ are the maximum numbers of the four components of ARGB, respectively, in the specified color space, and the maximum centroid distance $MD_{centroid}=\sqrt{w^2+h^2}$, where w and h are the width and height of the resized images, respectively. The normalized color distance ND_{color} is defined as

$$ND_{color}(dc_i, dc_j) = \frac{\sqrt{(dc_i-dc_j)\times(dc_i-dc_j)^T}}{MD_{color}}. \quad (8)$$

The normalized centroid distance $ND_{centroid}$ is defined as

$$ND_{centroid}(C_{dc_i}, C_{dc_j}) = \frac{\sqrt{(C_{dc_i}-C_{dc_i})\times(C_{dc_i}-C_{dc_i})^T}}{MD_{centroid}}. \quad (9)$$

The normalized feature distance between φ_i and φ_j is defined as

$$ND_{feature}(\varphi_i,\varphi_j) = p\cdot ND_{color}(dc_i, dc_j) \\ + q\cdot ND_{centroid}(C_{dc_i}, C_{dc_j}). \quad (10)$$

So far, $\boldsymbol{D}=[d_{ij}]$, where $d_{ij}=ND_{feature}(\varphi_i,\varphi_j)$ can be calculated before performing EMD calculation.

Suppose we have signature $S_{s,a}$ and signature $S_{s,b}$, where $S_{s,a}$ has m features and $S_{s,b}$ has n features. The flow matrix $\boldsymbol{F}_{ab}=[f_{ij}]$ $(1\le i\le m, 1\le j\le n)$ can be calculated through linear programming and the EMD between $S_{s,a}$ and $S_{s,b}$ can be calculated as:

$$EMD(S_{s,a}, S_{s,b}, D) = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}\cdot d_{ij}}{\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}}. \quad (11)$$

Experiments show that the linear programming within 100 features for each signature can normally be calculated in 100 iterations when the "Epsilon" for indicating optimization is set as 1e-6, which is sufficiently precise to calculate the EMD of two signatures. Since $d_{ij}=ND_{feature}(\varphi_i,\varphi_j)\in[0,1]$, $EMD(S_{s,a}, S_{s,b}, D)\in[0,1]$. If $EMD(S_{s,a}, S_{s,b}, D)=0$, the two images are completely identical, and if $EMD(S_{s,a}, S_{s,b}, D)=1$, the two images

are completely different. We define EMD-based visual similarity of two images as:

$$VS(S_{s,a}, S_{s,b}) = 1 - [EMD(S_{s,a}, S_{s,b}, D)]^{\alpha}, \quad (12)$$

where $\alpha \in (0, +\infty)$ is the amplifier of visual similarity. We use α to make visual similarity to be better distributed in (0,1) rather than too dense at either side without affecting the ranking relationship of the visual similarity values of Web pages.

5 Classification

We use a special threshold for each given protected Web page to classify a Web page to be a phishing Web page or a normal one. If we have a set of protected Web pages, we have to calculate the threshold vector for them first. When a suspected Web page comes, we calculate its visual similarity to all of the protected ones and determine if it is a phishing one to any of them. The threshold vector is represented as:

$$T = < T_1, T_2, \ldots, T_{N_{protected}} >,$$

where $T_i (1 \leq i \leq N_{protected})$ denotes the threshold of the ith protected Web page. $T_i (1 \leq i \leq N_{protected})$ is defined as:

$$T_i = \underset{t \in VSS_i}{argmin}(MissClassification(t)) - \delta, \quad (13)$$

where VSS_i is the historic visual similarity set of (i.e., contains the similarity values of the historically tested Web pages to) the ith protected Web page, and $MissClassification(t)$ is the number of misclassified Web pages in case we use t as the threshold. There are two types of misclassifications: 1) false alarm: the visual similarity is larger than or equal to t but, in fact, the Web page is not a phishing Web page (false positive) and 2) missing: the visual similarity is less than t but, in fact, the Web page is a phishing one (false negative). t should be selected as small as possible without increasing the misclassified number. We use the dynamic programming method to calculate t. Each phishing detection record in VSS_i correlates to two accessory parameters, the false alarm number (fa) and false negative (fn). They denote when we use the EMD of a training record as the threshold, how many false alarms and false negatives it will yield. We simply choose the t that can minimize $fa + fn$. In case more than one t can achieve the same minimal $fa + fn$, we choose the smaller one. When a new training record arrives, we need to set new fa and fn for this record, adjust all other accessory parameters in VSS_i, and choose the EMD of the record that can minimize $fa + fn$ again. Obviously, the new t can simply be found near the original one. δ is a slack constant and we use it to decrease missing cases by increasing false alarms because false negative is much more harmful than false alarms in antiphishing systems. δ can be chosen empirically from 0 to 0.1 under our observation from large-scale experiments and more detail is discussed in Section 6.

When a suspected Web page comes, we calculate the visual similarity vector which can be represented as

$$VS = < vs_1, vs_2, \ldots, vs_{N_{protected}} >,$$

where $vs_i (1 \leq i \leq N_{protected})$ denotes the visual similarity of this Web page to the ith protected Web page. We calculate the classification result using the following equation:

$$IsPhishing(VS) = \begin{cases} 1 & if\ max(VS - T) \geq 0 \\ 0 & if\ max(VS - T) < 0, \end{cases} \quad (14)$$

where $max(VS - T)$ denotes the maximum factor in $VS - T$. $IsPhishing(VS) = 1$ indicates that the Web page is a phishing Web page, while $IsPhishing(VS) = 0$ indicates that the Web page is normal.

6 Experiments

We carried out large-scale experiments to show the performance of our EMD-based phishing detection approach. We have used 10,272 homepage URLs retrieved from Google with 26 keywords as queries: bank, biology, car, Chinese, company, computer, English, entertainment, government, health, Hong Kong, house, Linux, money, movie, network, phishing, regional, research, science, spam, sport, television, university, Web, and Windows. We use each word to collect up to 1,000 homepage URLs. Duplicated URLs are removed to keep each URL unique. In addition to these Web pages, we have nine phishing Web pages collected from real phishing attack cases. The 10,281 (10,272+9) Web pages form the Suspected Web Page Set in our experiments. The Protected Web Page Set contains eight real Web pages attacked by the nine phishing Web pages. (These data sets are available on our Web site [9]).

We empirically set the parameters $w = h = 100$, $\alpha = 0.5$, $|S_s| = 20$, $p = q = 0.5$, and $\text{CDF} = 32$ in our experiments. This configuration provides satisfactory recognition of phishing Web pages and acceptable computation time at the same time, as we can see from the experiments (see Section 6.3 for details).

We first demonstrate in Section 6.1 the performance of our method using dynamically supervised training starting with a zero threshold vector (the similarity thresholds for all protected Web pages are 0). Then, we demonstrate in Section 6.2 the performance of our method with a trained threshold vector (or when the trained threshold vector is given). Finally, we discuss the parameter tuning process in Section 6.3.

6.1 Experiment Result with Dynamically Supervised Training

Assume our antiphishing system starts without any pretraining, i.e., all elements in the threshold vector are set to 0 initially. It is trained while it is running. When a suspected Web page comes, the threshold vector is dynamically adjusted using the threshold calculation method addressed in Section 5. The suspected Web pages in the Suspected Web Page Set are randomly ordered and

TABLE 1
Phishing Detection Performance of our Proposed Method (Using the Image-Based EMD)

δ	Total Alarm Number	Correct Number
0	42	7
0.005	73	8
0.01	134	8
0.015	229	8
0.02	403	9
0.025	693	9

sent to our system one by one. We record the classification results in each step. Table 1 shows the statistic result of the experiment after we test all the 10,281 suspected Web pages. Most of the phishing Web pages (seven out of nine) are detected when $\delta = 0$ and almost all phishing Web pages (eight out of nine) can be detected when $\delta = 0.005$. The last phishing Web page is difficult to detect because it is not visually similar to the real one at all, as shown in Fig. 2, which also shows that the most similar Web page of Real ICBC Asia under this measure among the 10,281 suspected Web pages is www.frlp.utn.edu.ar. Since our phishing detection method is based on visual similarity, it cannot detect those phishing Web pages that look different from the real one. However, from our former experiences and investigation, almost all phishers have tried their best to make the appearance of their phishing Web pages similar to the real ones. The reason could be that the phishers do not want to decrease their potential victims' response rate.

To demonstrate the advantage of our method, we experimentally compare it with two additional visual methods for phishing detection: 1) HTML/DOM-based EMD and 2) Region Matching, which is the method addressed in [17], [18], [19].

Method 1) segments the given Web page into blocks. Each block is represented with a DOM-based signature (size, position, foreground color, and background color, etc.). The Web page is represented with a DOM-based signature too by combining the block level signatures, which are used to calculate the EMD-based similarity. Table 2 shows the statistic result of this experiment.

Method 2) measures the visual similarity of Web pages in terms of similar key regions/blocks, similar page layout, and similar styles (e.g., font family, size, decoration, and even spacing). In this method, a Web page is finally reported as a phishing suspect for human confirmation if the visual similarity is higher than its corresponding preset

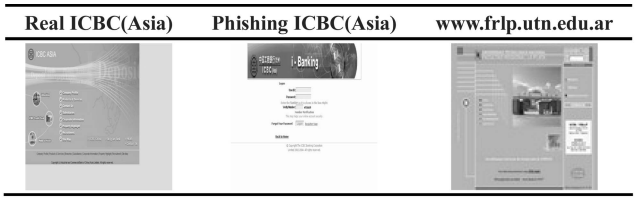


Fig. 2. Visual comparison of real and phishing ICBC (Asia) Web pages.

TABLE 2
Phishing Detection Performance of Method 1) (Using the HTML/DOM-Based EMD)

δ	Total Alarm Number	Correct Number
0	61	7
0.005	99	7
0.01	153	7
0.015	257	7
0.02	393	7
0.025	572	7
0.03	857	8
0.035	1244	9
0.04	1925	9

threshold. In their paper [17], [18], [19], the threshold is fixed for all protected Web pages. We carried out the same large-scale experiments using the 10,281 suspected Web pages and show its performance statistics by varying the threshold, as shown in Table 3.

In these three experiments, the Phishing ICBC Asia Web page is always the most difficult one to be detected. Considering only the other eight reasonable phishing Web pages, which are visually similar to their real ones, our proposed method performs the best. It is superior to both method 1) and 2). While our method can recognize these eight visually similar phishing Web pages at $\delta = 0.005$ by giving 65 (73-8) false alarms, method 1) can do this in the best case at $\delta = 0.03$ but results in 849 (857-8) false alarms

TABLE 3
Phishing Detection Performance of Method 2) (Using the Method in [17], [18], [19])

Threshold	Total Alarm Number	Correct Number
0.985	2	2
0.97	2	2
0.955	2	2
0.94	3	3
0.925	3	3
0.91	3	3
0.895	3	3
0.88	3	3
0.865	3	3
0.85	7	3
0.835	13	3
0.82	30	4
0.805	71	5
0.79	170	6
0.775	376	6
0.76	705	8
0.745	1230	8
0.73	1974	8
0.715	3029	9
0.7	4617	9

TABLE 4
Thresholds for Protected Web Page Set Trained with the 1,009 (1,000+9) Training Web Pages

Protected Web Page	Threshold
real-Bank of Oklahoma - Online	0.8469
real-eBay1	0.9434
real-eBay2	0.9493
real-ICBC(Asia)	0.7385
real-Key Bank	0.9323
real-US Bank	0.9573
real-Washington Mutual	0.8541
real-Wells Fargo Sign On	0.9255

and method 2) can do this at $\text{threshold} = 0.76$ but produces 697 (705-8) false alarms. In addition, we recall that the parameter needs to be adjusted in our method and method 1) is δ; however, the parameter needs to be adjusted in method 2) is the threshold. Hence, we use different parameters to evaluate the performances of the three methods.

6.2 Experiment Result with Trained Threshold Vector

In this experiment, we train the threshold vector using a training data set. From the 10,272 randomly collected Web pages, we select 1,000 Web pages and combine them with the nine phishing Web pages together to form the training data set (containing 1,009 Web pages). We use the training data set to calculate the thresholds for the eight protected Web pages. The training result is listed in Table 4. We use the other 9,272 (10,272-1,000) Web pages and combine them with the nine phishing Web pages to form the Suspected Web page Set. Table 5 shows the classification precision, phishing recalls, and false alarms of this Suspected Web page Set using these thresholds. There is only one missing case in this experiment due to the very reason that the real and phishing Web pages of ICBC (Asia) are not visually similar to each other. Hence, the reasonable upper bound of classification precision and phishing recall are 99.87 percent and 88.88 percent, respectively.

In practical applications, we are expecting to achieve better recall even though the precision and false alarm could be sacrificed a little (we treat the missing problem as more severe than the false alarm problem in antiphishing). We can look forward to reporting more phishing Web pages to achieve better recall rate. We use a slack constant δ to detect more potential phishing Web pages. Table 6 shows the classification precision, phishing recall, and false alarm values when we set $\delta = 0.005$. Fig. 3 shows the Web page of the "Real-Bank of Oklahoma" and one of its phishing Web pages detected.

TABLE 5
Classification Precision, Phishing Recall, and False Alarm List (Evaluated with $10,272 - 1,000 + 9 = 9,281$ Suspected Web Pages, $\delta = 0$)

Protected Web Page	Classification Precision	Phishing Recall	False Alarm
real-Bank of Oklahoma	9280/9281	1/1	1
real-eBay1	9280/9281	2/2	1
real-eBay2	9280/9281	1/1	1
real-ICBC(Asia)	9276/9281	0/1	5
real-Key Bank	9280/9281	1/1	1
real-US Bank	9280/9281	1/1	1
real-Washington Mutual	9280/9281	1/1	1
real-Wells Fargo	9280/9281	1/1	1
Overall	99.87%	88.88%	12

TABLE 6
Classification Precision, Phishing Recall, and False Alarm List (Evaluated with $10,272 - 1,000 + 9 = 9,281$ Suspected Web Pages, $\delta = 0.005$)

Protected Web Page	Classification Precision	Phishing Recall	False Alarm
real-Bank of Oklahoma	9277/9281	1/1	4
real-eBay1	9278/9281	2/2	3
real-eBay2	9278/9281	1/1	2
real-ICBC(Asia)	9275/9281	0/1	5
real-KeyBank	9279/9281	1/1	2
real-US Bank	9275/9281	1/1	6
real-Washington Mutual	9279/9281	1/1	2
real-Wells Fargo	9277/9281	1/1	4
Overall	99.70%	88.88%	28

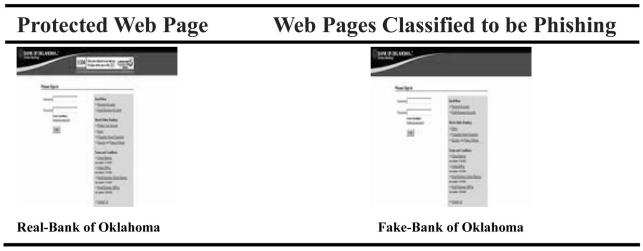


Fig. 3. Correct detection result for the Web page of the "Real-Bank of Oklahoma."

Fig. 4 shows four false alarm examples of the "Real-Bank of Oklahoma" when $\delta = 0.005$. The first three false alarm Web pages look indeed very similar to the Web page of the "Real-Bank of Oklahoma." Hence, it is reasonable and worthwhile to classify them as phishing Web pages for further examination. The fourth one does not look similar to humans but it is also classified as a phishing under $\delta = 0.005$. However, we consider it as a necessary sacrifice to reduce false negative possibilities.

6.3 Impact of the Parameters and Parameter Tuning

The results in Section 6.1 and 6.2 are based on the best parameter configuration we obtained from empirical evaluation. These experiments also show that the system performance is quite good under these parameter settings. We found that the global optimal parameters are quite hard to find from experiments because they need a huge amount of calculation, and we are also not expecting to improve the performance a lot through calculating the global optimal parameters. However, it could be more interesting to find out whether these empirical parameter settings can achieve a roughly local optimal performance. We did experiments to tune the parameters (w, h; $|S_s|$; p, q; and CDF) to see if we can find better parameter settings, which can also assess the goodness of these parameter settings.

The tuning approach is comprised of four parts: w, h; $|S_s|$; p, q; and CDF. As α is only an amplifier and it does not affect Web page similarity ranking and classification, we do not need to tune α. We simply set $\alpha = 0.5$. Each tuning is based on the start point of the default parameter setting: $w = h = 100$, $|S_s| = 20$, $p = q = 0.5$, and CDF $= 32$. When we tune one part, the other parts are fixed as the above setting. We use human vision to determine whether two Web pages are similar when building the ground truth data set. We first find out all similar Web pages for each protected Web page manually. The similar Web pages we manually determined for each protected Web page form a ground truth group data set for benchmarking. There are a total of 255 Web pages determined in these eight groups. Table 7 shows the number of ground-truth similar Web pages for each protected Web page. We take $N_{sample} \in \{5, 10, \ldots, 50\}$ as the sample number for each protected Web page. From the 10,281 suspected Web pages, the N_{sample} most similar Web pages, which are evaluated by our method, to each protected Web page are automatically checked for correctness according to our ground truth data sets. If a Web page in the N_{sample} collected Web pages is in the corresponding ground truth group, we count it as a correctly detected similar Web page. We count the total number of correctly detected Web pages for all of the eight protected Web pages and use it as the vertical axes of Fig. 5, Fig. 6, Fig. 7, and Fig. 8. In these figures, each line represents a different N_{sample} value and the horizontal axes show the variations of parameters we want to tune.

TABLE 7
Number of Ground Truth Web Pages for Each Protected Web Page in our Ground Truth Data Sets

Protected Web Page	Sample Number
real-Bank of Oklahoma - Online	85
real-eBay1	8
real-eBay2	19
real-ICBC(Asia)	30
real-Key Bank	14
real-US Bank	47
real-Washington Mutual	42
real-Wells Fargo Sign On	10
Total	255

6.3.1 *Tuning w and h*

We have four configuration options ($w = h = 10$, $10\sqrt{10}$, 100, *and* $100\sqrt{100}$) to tune w and h. As shown in Fig. 5, the ranking performance of our approach reaches a peak at $w \times h = 100 \times 100$. Although the CUP time is proportional to $w \times h$, the experiment shows that one pair of visual similarity can be computed in less than 0.1 seconds and $w \times h = 100 \times 100$ is a reasonable choice. Hence, it is reasonable to determine $w = h = 100$. However, it does

Fig. 4. Examples of wrong classification for the "Real-Bank of Oklahoma."

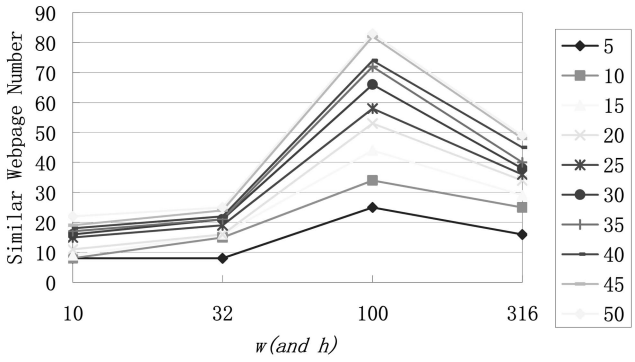


Fig. 5. Performance variation with w and h.

not mean $w \times h = 100 \times 100$ is optimal. $w = h = 100$ only indicates the roughly local optimal configuration of w and h. The same also applies to Fig. 6, Fig. 7, and Fig. 8.

6.3.2 Tuning $|S_s|$

We use six configuration options ($|S_s| = 5$, 10, 15, 20, 25, and 30) to tune $|S_s|$. As shown in Fig. 6, the ranking performance of our approach reaches a peak at $|S_s| = 20$. Although it does not lose much performance when $|S_s|$ is larger than 20, the time performance is worse since, when we use a bigger color number to compute visual similarity, and more CPU time is consumed. $|S_s|$ should be as low as possible if the classification performance can be guaranteed. Thus, $|S_s| = 20$ is a reasonable choice.

6.3.3 Tuning p and q

We use 11 configuration options ($p : q = 0 : 1, 0.1 : 0.9, 0.2 : 0.8, \ldots, 0.9 : 0.1, 1 : 0$) to tune p and q. As shown in Fig. 7, we could see that the ranking performance of our approach does not vary a lot with p and q. We think the reason is that color and color distribution in an image have similar impact on human eye perception. When p ranges from 0.3 to 0.7, the ranking performance is relatively high. It is reasonable if we use $p = q = 0.5$. As we can see, the proportion of p and q does not affect the performance much. Hence, we would also suggest simply using $p = 1$ *such that* the calculation of $ND_{centroid}$ can be avoided and the computing time can be saved.

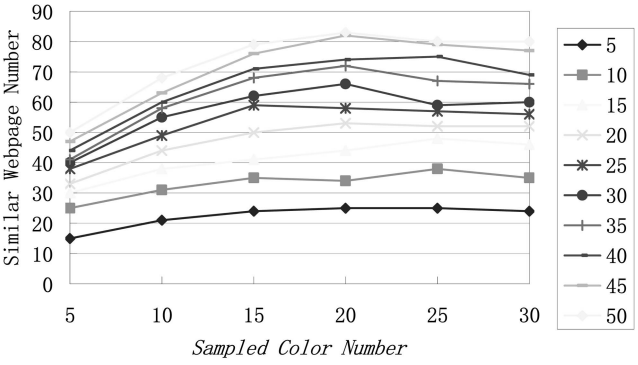


Fig. 6. Performance variation with $|S_s|$.

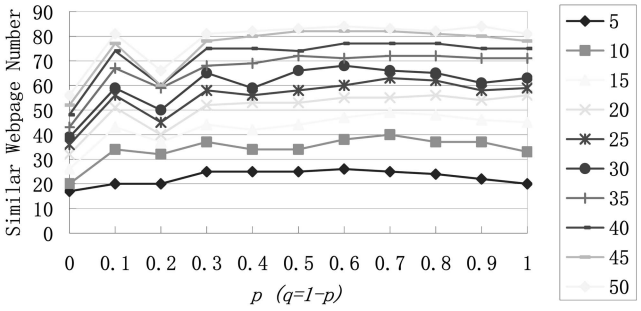


Fig. 7. Performance variation with p and q.

6.3.4 Tuning CDF

We use eight configuration options (CDF = 8, 16, 24, 32, 40, 48, 56, and 64) to tune CDF. As shown in Fig. 8, we can set CDF = 32 to obtain the best performance.

Finally, we decide to use the following parameter configuration: $w = h = 100$, $\alpha = 0.5$, $|S_s| = 20$, $p = q = 0.5$, and CDF = 32. In the aspect of time efficiency, each EMD-based visual similarity of two Web pages can be calculated in 0.02 second using an ordinary PC (with a single Intel Xeon 3.0G CPU, 4 Hyper Threads, and 512M RAM) with this parameter configuration, which is sufficiently fast for online phishing detection.

7 The Antiphishing System

We built an antiphishing system using the proposed approach. The system can automatically detect potential phishing Web pages by comparing their similarities to the protected Web pages. To our knowledge, almost all phishings start from sending phishing e-mails to Internet users. Spoofing contents are written in this kind of e-mail to make people eager to access their Web site and fill in personal information. We built the antiphishing engine into the Antiphishing Proxy which filters all traffics going through the e-mail server. We also built and deployed an Antiphishing Server, which acts like antivirus database servers. The Antiphishing Proxy keeps the phishing definitions updated from the Antiphishing Database Server. Fig. 9 shows the architecture of our system. The Antiphishing Database Server is the center for registration of legitimated Web sites which want protection, and maintains a phishing link list. It also acts logically as a part of the

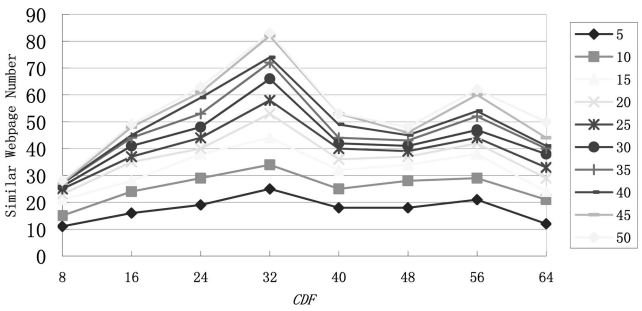


Fig. 8. Performance variation with CDF.

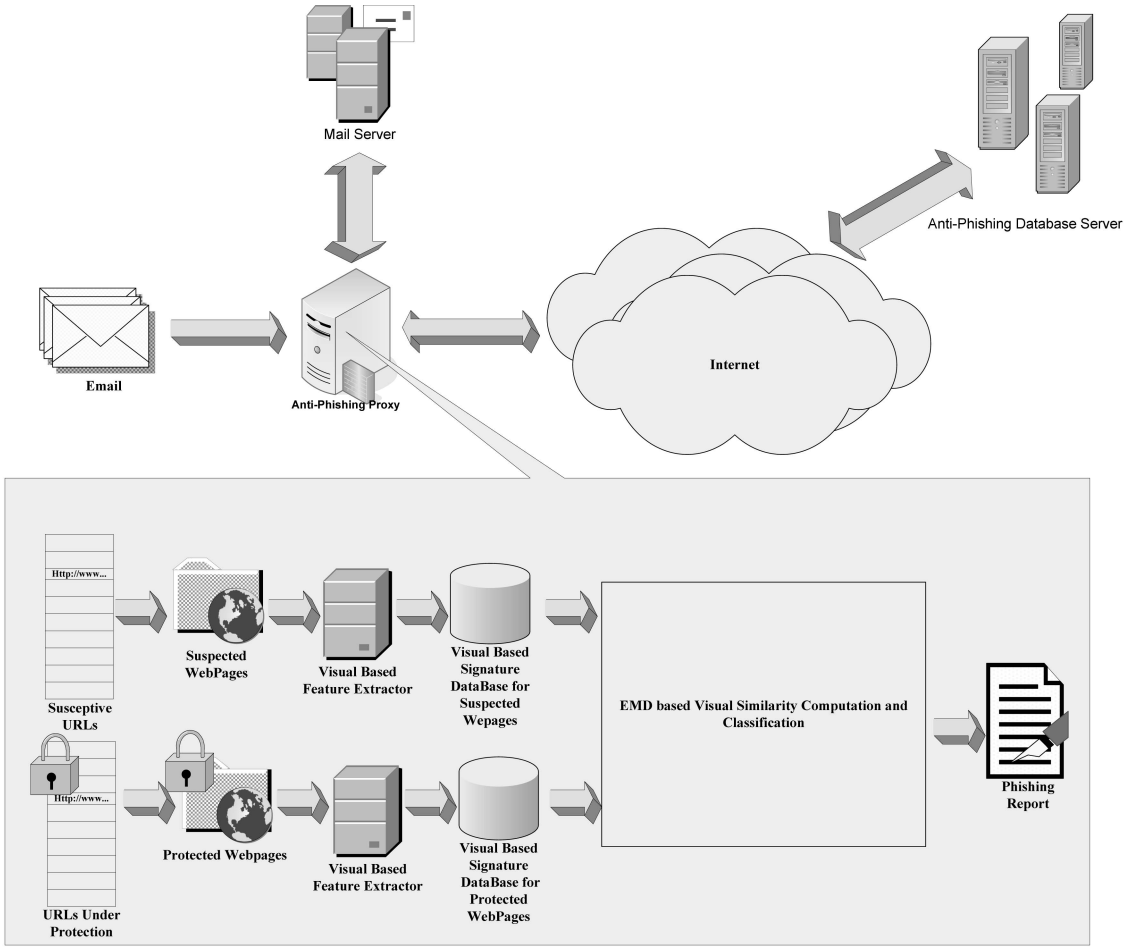


Fig. 9. Architecture of the antiphishing system.

Antiphishing Proxy to preprocess protected Web pages, such that this design makes good system scalability.

The registered legitimate Web pages are preprocessed in advance, and their signatures are saved in the database. The owner of the protected Web page can specify a set of sensitive keywords (e.g., "eBay" for www.ebay.com) during registration to the Antiphishing Database Server, which are used for checking the e-mails. Antiphishing Proxy synchronizes the database from the Antiphishing Database Server. If an e-mail contains any of these sensitive words, the URLs in this e-mail are parsed out. URLs that are in the protected list are deleted and the URLs that can be found from the phishing link list are reported immediately. The rest of the URLs are suspected ones, which are compared with the protected Web page associated with the sensitive word(s) using the proposed approach. If the visual similarity is larger than the threshold associated with the protected Web page, the suspected Web page is classified as phishing and an alert is reported. The system administrator can read the alert on behalf of the Antiphishing Database Server and make a final decision on whether the report is correct or not. The feedback from the administrator can help make a more accurate classification later. By doing so, a lot of human efforts can be saved for phishing monitoring and detection. Many phishing cases have been caught by our system and listed at [1].

8 Conclusions and Future Works

Phishing has become a severe problem of Internet security. We propose a phishing Web page detection method using the EMD-based visual similarity assessment. This approach works at the pixel level of Web pages rather than at the text level, which can detect phishing Web pages only if they are "visually similar" to the protected ones without considering the similarity of the source codes. Experiments also show that our method can achieve satisfying classification precision and phishing recall and the time efficiency of computation is acceptable for online use. An antiphishing system is built up on a mail server, which could be a prototype for industrial application.

However, our method is a visual method and assumes the phishing Web pages are visually similar to their attacking targets. The method could not detect those which are not visually similar. In order to address this problem, we will consider using textual features together with the visual features in our future work. There are also other potentially promising works to do for further exploration. The approach proposed in this paper is an important component in the antiphishing system. However, other techniques, e.g., OCR, Web page semantic structure analysis, natural language level semantic analysis, etc., could also be explored to further improve the detection accuracy.

Although most phishers will make the phishing Web page looks as similar as possible to the real one, our proposed system assesses the complete Web page rather than assessing a part of it. If a phisher makes a partially similar Web page to the real one, our system may fail. Hence, our system can be further improved in this aspect.

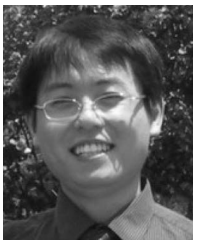
The usage of the method proposed in this paper may not just be limited to the server sides. We are also working on developing a client-side application, SiteWatcher Client, which can be installed by individual users for phishing detections (a trial version can be download at [1]). SiteWatcher Client works in a similar way to antivirus applications. It can periodically update the phishing database from SiteWatcher Server. It will also have a function to report new discovered phishing links to SiteWatcher Server. SiteWatcher will consider adding reported phishing links into the phishing database. SiteWatcher Client can monitor the TIC/IP packets a PC received and issue alerts once possible phishing URLs are detected.

ACKNOWLEDGMENTS

The work described in this paper was fully supported by grants from the City University of Hong Kong (Project No. 7001771 and 7001975) and the China Semantic Grid Research Plan (National Grand Fundamental Research 973 Program, Project No. 2003CB317002).

REFERENCES

[1] Anti-Phishing Group of the City University of Hong Kong, http://antiphishing.cs.cityu.edu.hk, 2005.
[2] Anti-Phishing Working Group, http://www.antiphishing.org, 2005.
[3] A. Broder, S. Glassman, M. Manasse, and G. Zweig, "Syntactic Clustering of the Web," *Proc. Sixth Int'l World Wide Web Conf.*, pp. 391-404, 1997.
[4] Y. Chen, W.Y. Ma, and H.J. Zhang, "Detecting Web Page Structure for Adaptive Viewing on Small Form Factor Devices," *Proc. 12th Int'l Conf. World Wide Web*, pp. 225-233, 2003.
[5] A. Chowdhury, O. Frieder, D. Grossman, and M. McCabe, "Collection Statistics for Fast Duplicate Document Detection," *ACM Trans. Information Systems*, vol. 20, no. 2, pp. 171-191, 2002.
[6] S. Cohen and L. Guibas, "The Earth Mover's Distance under Transformation Sets," *Proc. IEEE Int'l Conf. Computer Vision*, vol. 2, pp. 1076-1083, 1999.
[7] R. Dhamija and J.D. Tygar, "The Battle Against Phishing: Dynamic Security Skins," *Proc. Symp. Usable Privacy and Security*, 2005.
[8] A.Y. Fu, X. Deng, and W. Liu, "A Potential IRI Based Phishing Strategy," *Proc. Sixth Int'l Conf. Web Information Systems Eng. (WISE '05)*, pp. 618-619, Nov. 2005.
[9] A.Y. Fu, www.cs.cityu.edu.hk/~anthony/AntiPhishing, 2005.
[10] K. Grauman and T. Darrell, "Fast Contour Matching Using Approximate Earth Mover's Distance," *Proc. 2004 IEEE CS Conf. Computer Vision and Pattern Recognition*, vol. 1, pp. 220-227, 2004.
[11] X.D. Gu, J.L. Chen, W.Y. Ma, and G.L. Chen, "Visual Based Content Understanding towards Web Adaptation," *Proc. Second Int'l Conf. Adaptive Hypermedia and Adaptive Web-Based Systems*, pp. 29-31, 2002.
[12] F.S. Hillier and G.J. Liberman, *Introduction to Mathematical Programming*. McGraw-Hill, 1990.
[13] F.L. Hitchcock, "The Distribution of a Product from Several Sources to Numerous Localities," *J. Math. Physics*, vol. 20, pp. 224-230, 1941.
[14] T.C. Hoad and J. Zobel, "Methods for Identifying Versioned and Plagiarized Documents," *J. Am. Soc. Information Science and Technology*, vol. 54, no. 3, pp. 203-215, 2003.
[15] C.R. John, *The Image Processing Handbook*, second ed. CRC Press, 1995.
[16] E. Levina and P. Bickel, "The Earth Mover's Distance is the Mallows Distance: Some Insights from Statistics," *Proc. IEEE Int'l Conf. Computer Vision*, vol. 2, 2001.
[17] W. Liu, X. Deng, G. Huang, and A.Y. Fu, "An Anti-Phishing Strategy Based on Visual Similarity Assessment," *IEEE Internet Computing*, vol. 10, no. 2, pp. 58-65, 2006.
[18] W. Liu, G. Huang, X. Liu, M. Zhang, and X. Deng, "Detection of Phishing Web Pages Based on Visual Similarity," *Proc. 14th Int'l World Wide Web Conf.*, pp. 1060-1061, 2005.
[19] W. Liu, G. Huang, X. Liu, M. Zhang, and X. Deng, "Phishing Web Page Detection," *Proc. Eighth Int'l Conf. Documents Analysis and Recognition*, pp. 560-564, 2005.
[20] T. Nanno, S. Saito, and M. Okumura, "Structuring Web Pages Based on Repetition of Elements," *Proc. Seventh Int'l Conf. Document Analysis and Recognition*, 2003.
[21] Netscape Corp., The SSL Protocol, http://wp.netscape.com/eng/ssl3/, 2005.
[22] Y. Rubner, C. Tomasi, and L.J. Guibas, *"The Earth Mover's Distance as a Metric for Image Retrieval,"* Technical Report STAN-CS-TN-98-86, Dept. of Computer Science, Stanford Univ., 1998.
[23] Y. Rubner, C. Tomasi, and L.J. Guibas, "A Metric for Distributions with Applications to Image Databases," *Proc. IEEE Int'l Conf. Computer Vision*, pp. 59-66, 1998.
[24] G. Salton, A. Wong, and C.S. Yang, "A Vector Space Model for Information Retrieval," *J. Am. Soc. Information Science*, vol. 18, no. 11, pp. 613-620, 1975.
[25] L. Wood, *Document Object Model Level 1 Specification*, http://www.w3.org, 2005.
[26] M. Wu, R.C. Miller, and G. Little, "Web Wallet: Preventing Hishing Attacks by Revealing User Intentions," *Proc. Symp. Usable Privacy and Security*, 2006.
[27] S. Yu, D. Cai, J.R. Wen, and W.Y. Ma, "Improving Pseudo-Relevance Feedback in Web Information Retrieval Using Web Page Segmentation," *Proc. 14th Int'l Conf. World Wide Web*, pp. 11-18, 2003.



Anthony Y. Fu received the BSc degree in computer science from Tsinghua University and the PhD degree in computer science from the City University of Hong Kong. His research interests include artificial intelligence approaches to computer security and privacy, Web document analysis, information retrieval, and natural language processing. He is especially interested in devising ideas that can make an impact in the digital world. The goal is to release human labors in various areas and build ideas that can figure into the future of digital lives.

Liu Wenyin received the BEng and MEng degrees in computer science from Tsinghua University and the DSc degree from the Technion, Israel Institute of Technology. He is an assistant professor in the Computer Science Department at the City University of Hong Kong. His research interests include graphics recognition, engineering drawings recognition, and performance evaluation. In 2003, he was awarded the International Conference on Document Analysis and Recognition Outstanding Young Researcher Award by the International Association for Pattern Recognition. He is a senior member of the IEEE.

Xiaotie Deng received the BSc degree from Tsinghua University, Beijing, the MSc degree from Academia Sinica, Beijing, and the PhD degree from Stanford University. He is a professor of computer science at the City University of Hong Kong. His research interests include algorithmic game theory, Internet economics, online computing, and combinatorial optimization. He is a member of the ACM and a senior member of the IEEE.