Textual and Visual Content-Based Anti-Phishing: A Bayesian Approach

Haijun Zhang, Gang Liu, Tommy W. S. Chow, *Senior Member, IEEE*, and Wenyin Liu, *Senior Member, IEEE*

***Abstract*— A novel framework using a Bayesian approach for content-based phishing web page detection is presented. Our model takes into account textual and visual contents to measure the similarity between the protected web page and suspicious web pages. A text classifier, an image classifier, and an algorithm fusing the results from classifiers are introduced. An outstanding feature of this paper is the exploration of a Bayesian model to estimate the matching threshold. This is required in the classifier for determining the class of the web page and identifying whether the web page is phishing or not. In the text classifier, the naive Bayes rule is used to calculate the probability that a web page is phishing. In the image classifier, the earth mover's distance is employed to measure the visual similarity, and our Bayesian model is designed to determine the threshold. In the data fusion algorithm, the Bayes theory is used to synthesize the classification results from textual and visual content. The effectiveness of our proposed approach was examined in a large-scale dataset collected from real phishing cases. Experimental results demonstrated that the text classifier and the image classifier we designed deliver promising results, the fusion algorithm outperforms either of the individual classifiers, and our model can be adapted to different phishing cases.**

***Index Terms*— Bayes theory, classifier, data fusion, phishing detection, web page.**

I. Introduction

MALICIOUS people, also known as phishers, create phishing web pages, i.e., forgeries of real web pages, to steal individuals' personal information such as bank account, password, credit card number, and other financial data [1]–[3]. Unwary online users can be easily deceived by these phishing web pages because of their high similarities to the real ones. The Anti-Phishing Working Group [4] reported that there were at least 55 698 phishing attacks between January 1, 2009, and June 30, 2009. The latest statistics show that phishing remains a major criminal activity involving great losses of money and personal data.

Manuscript received September 6, 2010; revised June 30, 2011; accepted July 1, 2011. Date of publication August 4, 2011; date of current version October 5, 2011. This work was supported in part by the Natural Science Foundation of China under Grant 91024012.

H. Zhang was with the Department of Electronic Engineering, City University of Hong Kong, Kowloon, Hong Kong. He is now with the Department of Electrical and Computer Engineering, University of Windsor, Windsor, ON N9B3P4, Canada (e-mail: aarhzhang@gmail.com).

G. Liu and W. Liu are with the Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong (e-mail: gangliu@student.cityu.edu.hk; csliuwy@cityu.edu.hk).

T. W. S. Chow is with the Department of Electronic Engineering, City University of Hong Kong, Kowloon, Hong Kong (e-mail: eetchow@cityu.edu.hk).

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

Digital Object Identifier 10.1109/TNN.2011.2161999

Automatically detecting phishing web pages has attracted much attention from security and software providers, financial institutions, to academic researchers. Methods for detecting phishing web pages can be classified into industrial toolbar-based anti-phishing, user-interface-based anti-phishing, and web page content-based anti-phishing.

To date, techniques for phishing detection used by the industry mainly include authentication, filtering, attack tracing and analyzing, phishing report generating, and network law enforcement. These anti-phishing internet services are built into e-mail servers and web browsers and available as web browser toolbars (e.g., SpoofGuard Toolbar[1] [5], TrustWatch Toolbar[2], and Netcraft Anti-Phishing Toolbar[3]). These industrial services, however, do not efficiently thwart all phishing attacks. Wu *et al*. [6] conducted thorough study and analysis on the effectiveness of anti-phishing toolbars, which consist of three security toolbars and other mostly used browser security indicators. The study indicates that all examined toolbars in [6] were ineffective to prevent web pages from phishing attacks. Reports show that 20 out of 30 subjects were spoofed by at least one phishing attack, 85% of the spoofed subjects indicated that the websites look legitimate or exactly same as they visited before, and 40% of the spoofed subjects were tricked due to poorly designed web sites. Cranor *et al*. [7] performed another study on an evaluation of 10 anti-phishing tools. They indicated that only one tool could consistently detect more than 60% of phishing web sites without a high rate of false positives, whilst four tools were not able to recognize 50% of the tested web sites. Apart from these studies on the effectiveness of anti-phishing toolbars, Li and Helenius [8] investigated usability of five typical anti-phishing toolbars. They found that the main user interface of the toolbar, warnings, and help system are the three basic components that should be well designed. They also found that it is beneficial to apply whitelist and blacklist methods together. Also, due to the quality of the online traffic the applications from the anti-phishing client side should not rely merely on the Internet. Recently, Aburrous *et al*. [9] developed a resilient model by using fuzzy logic to quantify and qualify the website phishing characteristics with a layered structure and to study the influence of the phishing characteristics at different layers on the final phishing website rate.

Apart from the toolbar-based anti-phishing, typical techniques from the perspective of better user interfaces focus on helping users interact with a trusted web site. Dhamija and

[1] Available at http://crypto.stanford.edu/SpoofGuard.

[2] Available at http://geotrust.com.

[3] Available at http://toolbar.netcraft.com.

Tygar in [10] and Wu *et al.* in [11] designed prototype user interfaces, which force web page designers to follow certain paths to create web pages by adding either dynamic skin to web pages or sensitive information location attributes to HTML codes. In [12] and [13], a user's password is converted into a domain-specific password. This prevents phishers from obtaining the real password even if a user falls into a phishing web site.

Content-based anti-phishing, which is referred to as using the features of web pages, consists of surface level characteristics, textual content, and visual content. We clarify that the content of a web page we discuss here include the whole information of a web page such as a domain name, URL, hyperlinks, terms, images, and forms embedded in the web page. Surface-level characteristics have been commonly used by industrial toolbars to detect phishing. For example, the SpoofGuard makes use of inspecting the age of domain, well-known logos, URL, and links to acquire the characteristics of phishing web pages. Liu *et al.* [14] proposed the use of semantic link network (SLN) to automatically identify the phishing target of a given webpage. The method works by first finding the associated web pages of the given webpage and then constructing a SLN from all those web pages. A mechanism of reasoning on the SLN is exploited to identify the phishing target. Zhang *et al.* [15] developed a content-based approach, i.e., Carnegie Mellon Anti-phishing and Network Analysis Tool (CANTINA), for anti-phishing by employing the idea of robust hyperlinks [16]. Given a web page, this method first calculates the TF-IDF of each term, an algorithm usually used in information retrieval, generates a lexical signature[4] by selecting a few terms, supplies this signature to a search engine (e.g., Google), and then matches the domain name of current web page and several top search results to evaluate the current web page is legitimate or not. Another content-based technique, B-APT [17], is designed to identify phishing websites by using an open-source Bayesain filter on the basis of tokens which are extracted by a document object module (DOM) analyzer. The concept of visual approach to phishing detection was first introduced by Liu *et al.* [18]–[20]. This approach, which is oriented by the DOM-based [21] visual similarity of web pages, first decomposes the web pages (in HTML) into salient (visually distinguishable) block regions. The visual similarity between two web pages is then evaluated by three metrics, namely, block level similarity, layout similarity, and overall style similarity, which are based on the matching of the salient block regions [3]. Fu *et al.* [3] followed the overall strategy in [18]–[20], but proposed another method to calculate the visual similarity of web pages. They first converted HTML web pages into images and then employed the earth mover's distance (EMD) method [22] to calculate the similarity of the images. This approach only investigates phishing detection at the pixel level of web pages without considering the text level. Apart from these approaches to detect phishing web pages, content-based methods for detecting phishing emails have also been widely studied, especially using machine learning techniques. Chandrasekaran *et al.* [23] introduced a classification method based on structural characteristics of phishing emails, which employed information gain for feature selection and one-class support vector machines (SVM) for phishing classification. The performance of a number of widely used machine learning techniques in phishing detection was also compared [24]–[26]. It is noted that these methods were used to detect phishing web sites as well on the basis of only text features [24], [25] or to identify phishing e-mails based on a number of structural features such as age of domain name, presence of Javascript, presence of form tag, etc. [26].

The approach in this paper extends the method presented in [3] into a hybrid anti-phishing framework. This framework synthesizes multiple cues, i.e., textual content and visual content, from the given web page and automatically reports a phishing web page by using a text classifier, an image classifier, and a data fusion process of the classifiers. A Bayesian model is proposed to estimate the threshold, which is required in classifiers to determine the class of web page. We also develop a Bayesian approach to integrate the classification results from the textual and visual contents. The main contributions of this paper are threefold. First, we propose a text classifier using the naive Bayes rule [27], [28] for phishing detection. Second, we propose a Bayesian approach to estimate the threshold for either the text classifier or the image classifier such that classifiers enable to label a given web page as "phishing" or "normal." Third, we propose a novel Bayesian approach to fuse the classification results from the text classifier and the image classifier.

With respect to previous work, we clarify that our approach is most related to the content-based approaches such as CANTINA [15], visual similarity-based methods [3], [18]–[20], and machine learning techniques [24]–[26]. But the anti-phishing model proposed here is considerably different. In CANTINA [15], the formation of lexical signature is only based on several unique terms extracted from a given web page. The lexical signature is subsequently applied to the search engine. The generated lexical signature for the given web page matches with the domain name of billions of online web pages. The classification is based on the measurement from the PageRank [29] assumption. In our detection framework, the existence of a protected web page, i.e., a legitimate web page, needs to be determined in the first place. Thus, based on the statistics from the attack historical data of the protected web page, the system classifies a given web page into the corresponding category, i.e., either phishing or normal. In addition, we include the conditional probabilities of all words, while CANTINA essentially relies on identifying the most unique terms. Compared with the detection methods of [3], [18]–[20], we extend these methods into a hybrid anti-phishing framework, by taking additional content into account. Currently, we only include textual content as the additional content. Other surface level characteristics such as hyperlinks can also be easily combined into this framework. Here we only directly use the EMD method [3] to assess the visual similarity of web pages. The visual similarity measurements of [18]–[20] can also be easily used in this framework. In the text classifier, we at present use the naive Bayes rule to classify web pages.

[4]An example of such a lexical signature is available at http://xyz.com/nameofpage.html?lexical-signature='w1+w2+w3+w4+w5'.

Other machine learning techniques [24]–[26] such as random forests, neural networks, and SVMs, can also be examined in our framework but only at the text level. Furthermore, we determine the threshold used in classifiers by using the Bayesian approach. Our proposed fusion algorithm based on the Bayesian approach is also novel for phishing detection.

The remaining sections of this paper are organized as follows. In the following section, we provide an overview of our framework. In Section III, we introduce the text classifier based on the textual content of web pages. In Section IV, we introduce the image classifier and briefly describe the model to assess the similarity measurement of web pages proposed in [3]. In Section V, we introduce the Bayesian approach to estimate the threshold required in either the text classifier or the image classifier. In Section VI, we propose a novel fusion algorithm to combine the results from both classifiers. In Section VII, we perform extensive experiments on the evaluation of our proposed approach. In Section VIII, we end this paper with conclusions and future work propositions.

II. Overview of Our Framework

A. Framework of Our Approach

To summarize the whole content information of a web page, we divide the content representation into three categories.

1) Surface level content. "Surface level content" here is defined as the characteristics that are used by the users to access to a web page or to connect to other web pages. Such surface-level content consists of the domain name, URL, and hyperlinks which are involved in a given web page.
2) Textual content. "Textual content" in this paper is defined as the terms or words that appear in a given web page, except for the stop words (a set of common words like "a," "the," "this," etc.). We first separate the main text content from HTML tags and apply stemming [30] to each word. Stems are used as basic features instead of original words. For example, "program," "programs," and "programming" are stemmed into "program" and considered as the same word.
3) Visual content. "Visual content" refers to the characteristics with respect to the overall style, the layout, and the block regions including the logos, images, and forms. Visual content also can be further specified to the color of the web page background, the font size, the font style, the locations of images and logos, etc. In addition, the visual content is also user-dependent. On the other hand, we can consider the web page at the pixel level, i.e., an image that enables the total representation of the visual content of the web page.

In our framework, we only consider the textual and visual content of a web page, because the surface-level characteristics have been well embedded in the toolbars such as SpoofGuard, and the heuristics adopted in the toolbars also can be easily combined into our system. Thus, in this paper the content of a given web page is transformed into two categories, namely, the textual and the visual, which is addressed by the corresponding classifier. The proposed anti-phishing approach contains the following components.

1) A text classifier using the naive Bayes rules to handle the text content extracted from a given web page.
2) An image classifier using the EMD similarity assessment [3] to handle the pixel level content of a given web page that has been transformed into an image.
3) A Bayesian approach to estimate the threshold used in classifiers through offline training.
4) A data fusion algorithm to combine the results from the text classifier and the image classifier. The algorithm employs the Bayesian approach as well.

Fig. 1 illustrates an overview of our framework. The system includes a training section, which is to estimate the statistics of historical data (i.e., web page training set), and a testing section, which is to examine the incoming testing web pages. The statistics of the web page training set consists of the probabilities that a textual web page belongs to the categories (i.e., phishing and normal), the matching thresholds of classifiers, and the posterior probability of data fusion. Through the preprocessing, content representations, i.e., textual and visual, are rapidly extracted from a given testing web page. The text classifier is used to classify the given web page into the corresponding category based on the textual features. The image classifier is used to classify the given web page into the corresponding category based on the visual content. Then the fusion algorithm is used to combine the detection results delivered by the two classifiers. The detection results are eventually transmitted to the online users or the web browsers.

B. Implementation of an Anti-Phishing System

Phishers strive to mimic web pages of most well-known international banks, economic organizations, or other brands, because unwary online users may be easily scammed by these fake web pages. This motivates us to develop detection tools to protect the legitimate web pages from being frequently attacked. In this way, the web page that is designed for customers or users to access needs to be examined by matching the restored legitimate web pages. For example, a customer may usually use "eBay" to do shopping. Thus, we need protect the user from being phished by comparing the content of the given web pages with that of the real "eBay" web page. If both two web pages exhibit highly matched content, we claim the given web page is phishing. We can integrate our solution into a browser plug-in for the user to maintain and protect a list of frequently used web pages that need high security attention. Another alternative approach is to provide a class library application programming interfaces (APIs) for enterprises that build their own anti-phishing systems for detecting suspicious web pages. For example, "eBay" probably only cares about their own site, so it makes sense for them to detect fake versions of their own brand. On the other hand, our proposed approach is easy to be embedded into the current anti-phishing system [3], [18]–[20]. Since almost all phishings start from sending phishing emails to Internet users who are deceived by this kind of e-mails to access their fake web site and results in exposing their personal information [3], we can build

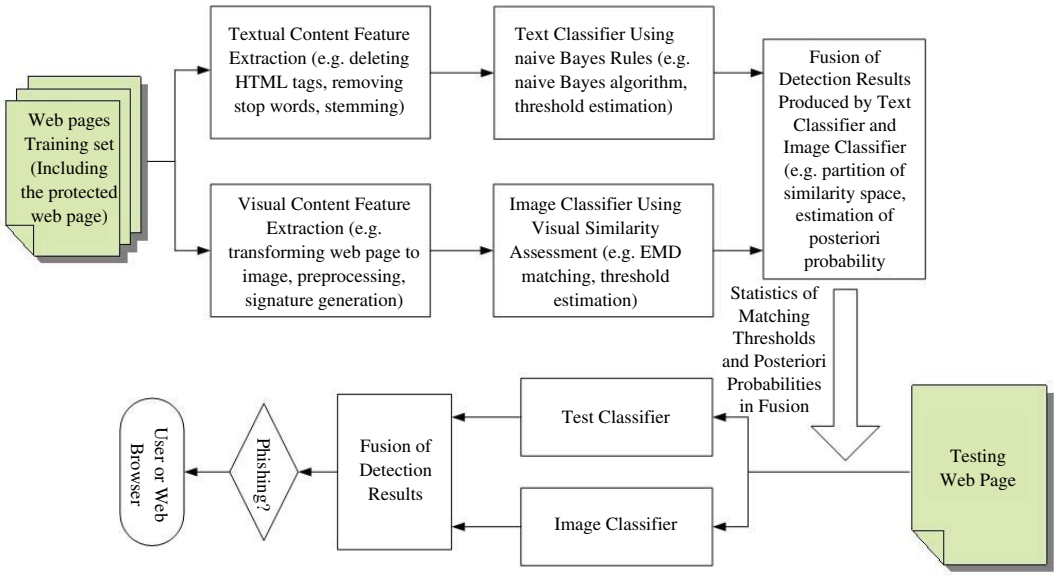

Fig. 1. Overview of the system framework.

an anti-phishing engine into the anti-phishing proxy to keep the phishing characteristics updated from the anti-phishing database server so as to filter all traffic going through the e-mail server. The anti-phishing database server is the center for registration of legitimate web sites that need protection. The registered legitimate web pages are preprocessed in advance. Their content features and historical anti-phishing statistics are extracted from the pages and saved in the database such that this design makes the system efficient and scalabile [3]. The overall implementation of such an anti-phishing system can be found in [3].

III. Text Classifier

A. Preprocessing

The main texts of a given web page are firstly separated from HTML tags. In order to form a histogram vector for each web page, we construct a word vocabulary. In this paper, we extract all the words from a given protected web page and apply stemming to each word. It is worth noting that using the naive word-based extraction may deliver more discriminative information than employing this stemming-based extraction. But we must point out that the naive word-based extraction will heavily increase the vocabulary size. For example, for "eBay" dataset (see Section VII), the vocabulary size with respect to the naive word-based extraction is 9570, while the vocabulary size is only 280 by using stemming. In addition, using stemming will deliver more robustness of detection, because phishers may manipulate the textual content through the change of tense and active to passive. The use of either the stemming-or naive word-based extraction depends on different objectives. For exact matching of textual content, we suggest using the naive word-based extraction, whilst for smaller vocabulary and more robust detection size we recommend using stemming. In this paper, stems are used as basic features instead of original words. We store the stemmed words to construct the vocabulary. Given a web page, we then form a histogram vector $(h_1, h_2, \ldots, h_n)$, where each component represents the term frequency (a term appears in the web page) and n denotes the total number of components in the vector. We explain three points here.

1) We do not extract words from all the web pages in a dataset to construct the vocabulary, because phishers usually only use the words from a targeted web page to scam unwary users.
2) For the sake of simplicity, we do not use any feature extraction algorithms in the process of vocabulary construction.
3) We do not take the semantic associations of web pages into account, because the sizes of most phishing web pages are small.

B. Bayesian Classifier

In this paper, we use the Bayes classifier to classify the text content of web pages. In the classifying process, the Bayes classifier outputs probabilities that a web page belongs to the corresponding categories. These probabilities also can be regarded as the similarities or dissimilarities that given web pages have with the protected web page. Let $G = \{g_1, g_2, \ldots, g_j, \ldots, g_d\}$ denote the set of web page categories, where d is the total number of categories. In fact, for anti-phishing problem only two categories are included: the phishing web page category g_1 and the normal web page category g_2. Given a variable vector $(v_1, v_2, \ldots, v_n)$ of a web page, the classifier is employed to determine the probability $P(g_j|v_1, v_2, \ldots, v_n)$ that the web page belongs to category g_j. Applying the Bayes rule, the posterior probability $P(g_j|v_1, v_2, \ldots, v_n)$ is calculated by

$$P(g_j|v_1, v_2, \ldots, v_n) = \frac{P(v_1, v_2, \ldots, v_n|g_j)P(g_j)}{P(v_1, v_2, \ldots, v_n)} \quad (1)$$

where the prior probability $P(g_j)$ is estimated by the frequency of the training samples belonging to category g_j.

It is difficult to directly estimate the conditional probability $P(v_1, v_2, \ldots, v_n|g_j)$, because the data samples are sparsely distributed in a high-dimensional space. However, since we ignore the semantic associations among terms, the naive Bayes classifier [27], [28] is used to handle the issue. Naive Bayesian theory assumes that all the components in the histogram vector are independent from one another. Thus the conditional probability is represented by

$$P(v_1, v_2, \ldots, v_n|g_j) = \prod_{i=1}^{n} P(v_i|g_j). \tag{2}$$

The joint probability $P(v_1, v_2, \ldots, v_n)$ is described by

$$P(v_1, v_2, \ldots, v_n) = \sum_{j=1}^{d} P(v_1, v_2, \ldots, v_n|g_j). \tag{3}$$

Then the posterior probability $P(g_j|v_1, v_2, \ldots, v_n)$ is transformed into

$$P(g_j|v_1, v_2, \ldots, v_n) = \frac{P(g_j)\prod_{i=1}^{n} P(v_i|g_j)}{\sum_{j=1}^{c}\prod_{i=1}^{n} P(v_i|g_j)}. \tag{4}$$

C. Implementation of Text Classifier

Let $C_j = \{c_{j,1}, c_{j,2}, \ldots, c_{j,K_j}\}$ be the set of training web pages belonging to category g_j, where K_j is the number of web pages in set C_j, and let $H_l = (h_{l,1}, h_{l,2}, \ldots, h_{l,n})$ ($l = 1, 2, \ldots, K_j$) denote the histogram vector of the l-th web page in C_j corresponding to the word vocabulary $(u_1, u_2, \ldots, u_n)$. Conditioning on category g_j, the estimation of the probability $P(u_i|g_j)$ of the i-th word in the vocabulary is given by

$$P(u_i|g_j) = \frac{1 + \sum_{l=1}^{K_j} h_{l,i}}{\sum_{i=1}^{n}\sum_{l=1}^{K_j} h_{l,i}} \tag{5}$$

and the estimation of the probability $P(g_j)$ is determined by

$$P(g_j) = \frac{K_j}{\sum_{j}^{d} K_j}. \tag{6}$$

Thus, given a testing web page T, the probability $P(g_j|T)$ that the web page T belongs to category g_j is calculated by

$$P(g_j|T) = \frac{P(g_j)\prod_{i=1}^{n} P(u_i|g_j)^{\frac{h_{i,T}}{R}}}{\sum_{s=1}^{d} P(g_s)\prod_{i=1}^{n} P(u_i|g_s)^{\frac{h_{i,T}}{R}}} \tag{7}$$

where, $h_{i,T}$ represents the frequency of the ith word appearing in the web page T, and R is the total number of words extracted from the protected web page. Here, the term R is used to enlarge the value of the term $P(u_i|g_j)^{h_{i,T}/R}$ such that the denominator of (7) will not be close to zero, because for most phishing cases the phishing web pages include much more term frequencies than the normal web pages. We then compare the probability $P(g_1|T)$ of the web page T belonging to the phishing category g_1 to a threshold θ_T which is estimated later by using the Bayesian theory (see Section V). If the probability $P(g_1|T)$ exceeds the threshold θ_T, the web page is classified as phishing, otherwise, the web page is classified as normal.

IV. Image Classifier

In reality, using only text content is insufficient to detect phishing web pages. This method will usually result in high false positives, because phishing web pages are highly similar to the targeted web pages not only in textual content but also in visual content such as famous logos, layout, and overall style. In this paper, we use the same approach as in [3] using the EMD to measure the visual similarity between an incoming web page and a protected web page. We briefly describe the preprocessing, feature representation, and distance measurement (i.e., applying the EMD) adopted in [3], as detailed description is beyond the scope of this paper.

A. Preprocessing and Feature Representation

First, we retrieve the suspected web pages and protected web pages from the web. Second, we generate their signatures, which are used for the calculation of the EMD between them. The graphic device interface API provided by the Microsoft IE browser is used to transform HTML and accessory files on the screen into web page images (in JPEG format). The images with the original sizes are processed into images with normalized sizes (e.g., 100×100). Here the Lanczos algorithm [31] is adopted to build the resized images, because the Lanczos algorithm has strong anti-aliasing abilities in the Fourier domain and is easily computed in spatial domain. Thus all the web page images are normalized into fixed-size square images. We use these normalized images to generate the signature of each web page [3].

A signature of an image, i.e., a feature vector, is used to represent the image. It consists of features and their corresponding weights. A feature includes two components: a degraded color and the centroid of its position distribution in the image. Let $F_\sigma = \{\sigma, C_\sigma\}$ be the feature, where σ represents the degraded color (i.e., a 4-tuple $< A, R, G, B >$, in which the components represent alpha, red, green, and blue, respectively), and C_σ represents the centroid of the degraded color. The calculation of the centroid is given by $C_\sigma = \sum_{i=1}^{N_\sigma}(c_{\sigma,i}/N_\sigma)$, where $c_{\sigma,i}$ is the coordinate of the ith pixel that has the degraded color σ, and N_σ is the total number of pixels that have the degraded color σ (i.e., the frequency). The weight corresponding to the feature F_σ is the color's frequency N_σ. Thus, a complete signature S is described as

$$S = \{(F_{\sigma_1}, N_{\sigma_1}), (F_{\sigma_2}, N_{\sigma_2}), \ldots, (F_{\sigma_N}, N_{\sigma_N})\} \tag{8}$$

where N is the total number of selected degraded colors. In this signature representation, the feature weighted units in S are ranked in the descending order of their weights, i.e., $N_{\sigma_i} \geq N_{\sigma_{i+1}}$ for $1 \leq i \leq N-1$ [3].

B. Distance Measurement

The EMD [22], [3] is adopted to measure the distance (or dissimilarity) of two web page images, because it supports many-to-many matching for feature distributions. Suppose we have two web page images a and b with signature S_a and S_b, respectively, where S_a has m feature units and S_b has n feature units. We first calculate the distance matrix $D = [d_{ij}]$

$(1 \leq i \leq m,\ 1 \leq j \leq n)$, where $d_{ij} = D_{norm}(F_{\sigma_i}, F_{\sigma_j})$. $D_{norm}(F_{\sigma_i}, F_{\sigma_j})$ is a normalized feature distance between feature F_{σ_i} and feature F_{σ_j}, which is defined by

$$D_{norm}(F_{\sigma_i}, F_{\sigma_j}) = \mu \cdot ||\sigma_i - \sigma_j|| + \eta \cdot ||C_{\sigma_i} - C_{\sigma_j}|| \quad (9)$$

where $\mu + \eta = 1$. Then the flow matrix $F_{ab} = [f_{ij}]$ is calculated through linear programming and the EMD between S_a and S_b is calculated by

$$EMD(S_a, S_b, D) = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij} \cdot d_{ij}}{\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}}. \quad (10)$$

We define the EMD-based visual similarity of two images as

$$S_{visual}(S_a, S_b) = 1 - (EMD(S_a, S_b, D))^{\alpha} \quad (11)$$

where $\alpha \in (0, +\infty)$ is the amplifier of visual similarity. If $S_{visual}(S_a, S_b) = 1$, the two images are completely identical, and if $S_{visual}(S_a, S_b) = 0$, the two images are completely different, because $EMD(S_a, S_b, D) \in [0, 1]$ [3].

It is true that the computational efficiency is a major concern of calculating the EMD between two images because a speedy response is always expected by users. The computation complexity of comparing two images can be solved in $O(m^3 \log(m))$ [21] if two images have the same number of feature units (i.e., $m = n$). There is no explicit expression for the case that two images have the different sizes of signatures. The average computational time of comparing a testing web page and a protected web page empirically studied in our system is around 1.43 s, which includes the time of transformation of the testing web page to an image, the image feature extraction, the signature generation, and the calculation of the EMD. Here, note that we do not consider the preprocessing time of the protected web pages, because we are able to preprocess these web pages offline and store the information of them into a database server as described in Section II-B. According to the 2s-rule [32], such time cost indicates that combining visual content into anti-phishing framework is applicable to the real phishing detection applications with an acceptable system response. On the other hand, the visual content matching method is not limited to the use of the EMD algorithm. Apart from the EMD, we can use other advanced image analysis methods such as [33]–[35]. These methods may further improve the time efficiency of the system. But it is beyond the scope of this paper, and we leave the investigation of other visual content matching algorithms to our other researchers' future work.

C. Implementation of Image Classifier

The image classifier is implemented by setting a threshold θ_V, which is later estimated in the subsequent section. If the visual similarity S_{visual} between a suspected web page and the protected web page exceeds the threshold θ_V, the web page is classified as phishing, otherwise, the web page is classified as normal. The overall implementation process of image classifier is summarized as follows.

Step 1: Obtain the images of a web pages from its URL and perform normalization.

Step 2: Generate visual signature of the input image including the color and coordinate features.

Step 3: Calculate the EMD and visual similarity between the input web page image and the protected web page image using (10) and (11).

Step 4: Classify the input web page into corresponding category according to the comparison of the visual similarity and the threshold θ_V.

V. Bayesian Threshold Estimation

A. Probabilistic Model

We use a threshold θ in either the text classifier or the image classifier to classify a web page to be a phishing web page or a normal one. One important issue is how to appropriately set this threshold such that the number of misclassified web pages can be minimized. Anti-phishing context includes two types of misclassifications [3]:

1) false alarm: the similarity S is larger than θ but, in fact, the web page is not a phishing web page (false positive);
2) false negative: the similarity S is smaller than or equal to θ but, in fact, the web page is a phishing one.

Here, the similarity S is the probability $P(g_1|T)$ of the web page T belonging to the phishing category g_1 in the text classifier (see Section III-C) or the visual similarity S_{visual} in the image classifier (see Section IV-C). Intuitively, we can directly estimate the threshold by counting the number of web pages mistakenly classified by the classifier in a large set of known training samples, which has been done in [3]. In this paper, we introduce a Bayesian approach to model the posterior probability of a phishing web page conditioning on a specified threshold, which is proved to equally minimize the number of misclassified web pages (see Appendix).

Let binary state random variable $E \in \{O, N\}$ be the event that a web page is a phishing or normal one and $s \in [0, 1]$ be the similarity variable. Motivated by [36], the desired Bayesian model to determine a posterior probability of a web page that is a phishing one conditioning on a threshold θ is given by

$$P(O|s > \theta) = \frac{P(O)P(s > \theta|O)}{P(s > \theta)}. \quad (12)$$

Since

$$P(s > \theta) = P(O)P(s > \theta|O) + P(N)P(s > \theta|N) \quad (13)$$

we obtain

$$P(O|s > \theta) = \frac{P(O)P(s > \theta|O)}{P(O)P(s > \theta|O) + P(N)P(s > \theta|N)}. \quad (14)$$

Thus, we can specify a threshold θ on the maximum of a posterior probability by (14). It has been proved that maximizing a posterior probability $P(O|s > \theta)$ conditioning on a threshold θ is equal to minimizinge the number of misclassified web pages (see Appendix for proof).

B. Implementation

Let $A = \{s \leq \theta\}$ be the event that the variable s associated with the similarity of a given web page and a protected web page, $X \in A$ be a continuous random variable, and $Y \in \{O, N\}$ be a binary random variable. Then we can estimate the conditional probabilities $P(s > \theta|O)$ and $P(s > \theta|N)$ in (14) by

$$P(s > \theta|O) = 1 - P(X \in A|Y = O) = 1 - \int_0^{\theta} f_{X,Y}(x, Y = O)dx \quad (15)$$

and

$$P(s > \theta|N) = 1 - P(X \in A|Y = N) = 1 - \int_0^{\theta} f_{X,Y}(x, Y = N)dx \quad (16)$$

respectively, where $f_{X,Y}(x, Y = O)$ and $f_{X,Y}(x, Y = N)$ represent density functions. In reality, however, it is difficult to determine the density functions unless we have enough statistics collected from massive data samples. In this paper, we collect the similarity measurements from known training sets to determine the threshold θ on the maximum of the posterior probability $P(O|s > \theta)$.

Let d_i denote the similarity between the ith web page (in the training set) and the protected web page. We first set $\theta = d_i$ and then go through the training set and obtain

$$P(s > \theta|O)_{\theta=d_i} = \frac{K(s > d_i, O)}{K(O)} \quad (17)$$

$$P(s > \theta|N)_{\theta=d_i} = \frac{K(s > d_i, N)}{K(N)} \quad (18)$$

$$P(O) = \frac{K(O)}{K_T} \quad (19)$$

$$P(N) = \frac{K(N)}{K_T} \quad (20)$$

where $K(s > d_i, O)$ and $K(s > d_i, N)$ denote the numbers of phishing and normal web pages, the similarities of which exceed d_i, respectively, $K(O)$ and $K(N)$ denote the number of phishing and normal web pages in the training set, respectively, and $K_T = K(O) + K(N)$ denotes the total number of web pages in the training set. We substitute (17)–(20) into (14) and obtain the posterior probability conditioning on $\theta = d_i$

$$P(O|s > \theta)_{\theta=d_i} = \frac{K(s > d_i, O)}{K(s > d_i, O) + K(s > d_i, N)}. \quad (21)$$

Then we select one of similarity measurements as the threshold θ that satisfies

$$\theta = \arg\max_{\hat{d}_i} \left(\frac{K(s > \hat{d}_i, O)}{K(s > \hat{d}_i, O) + K(s > \hat{d}_i, N)} \right) \left(\hat{d}_i \in \{\arg\max_{d_i} K(s > d_i, O)\} \right). \quad (22)$$

It is noted that the posterior probability $P(O|s > \theta)$ in (14) and (21) is limited by $P(O|s > \theta) \leq 1$. If $P(O|s > \theta) = 1$, then $P(s > \theta|N) = 0$, i.e., $K(s > d_i, N) = 0$ in (21). It indicates that we can select a threshold as large as possible to let $K(s > d_i, N) = 0$. But it is noted that $K(s > d_i, O)$ also decreases when the threshold saturates to 1 (the maximum value). We clarify the strong rationale of using our model through an example illustrated in Fig. 2. It shows an example including eight data samples, in which the first four samples are phishing web pages and the rest samples are normal web pages. In Fig. 2(a), the set of similarity measurements is $\Omega_a = \{0.8, 0.9, 0.7, 0.85, 0.33, 0.3, 0.6, 0.36\}$. In Fig. 2(b), the set of similarity measurements is $\Omega_b = \{0.8, 0.9, 0.55, 0.85, 0.33, 0.3, 0.6, 0.36\}$. As seen from Fig. 2(a), the samples can be nicely categorized by setting an appropriate threshold. In the estimation process, conditioning on certain thresholds $\theta \in \Omega_a$, we calculate the corresponding posterior probabilities using (21). The results are listed in Table I. It is observed that when $\theta = 0.6, 0.7, 0.8$, respectively, $P(O|s > \theta)$ are all equal to 1 given by $K(s > d_i, N) = 0$. Using our model in (22), $\theta = 0.6$ will be selected as the threshold, which is the optimal. In Fig. 2(b), $\theta = 0.36$ will be selected as the threshold. In Table II, it is noted that the posterior probability $P(O|s >$

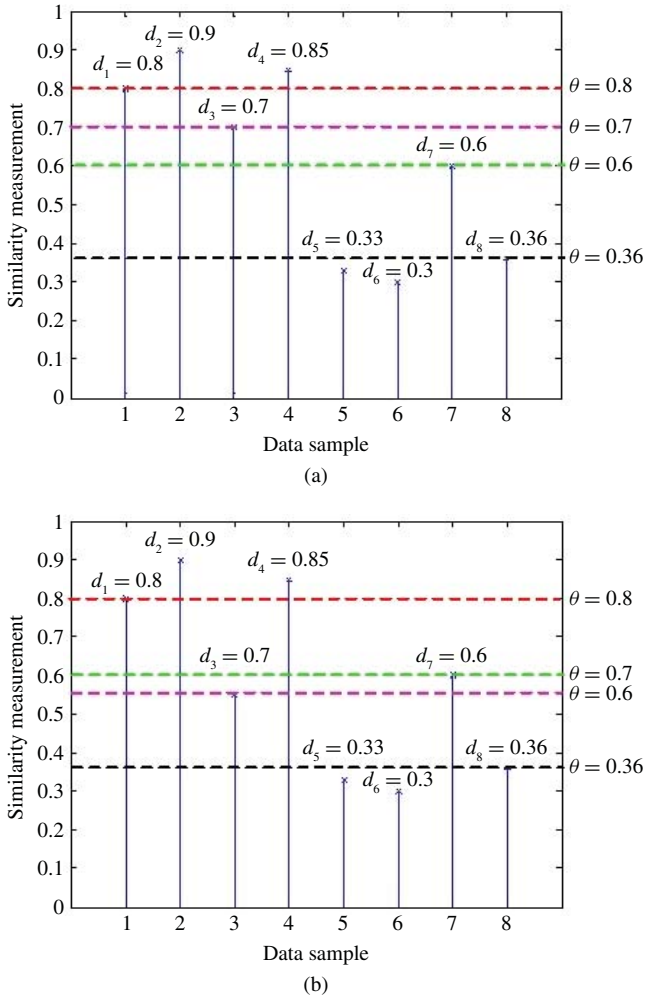

Fig. 2. Example of threshold estimation including eight data samples, in which the first four samples are phishing web pages and the rest are normal web pages, the set of similarity measurements in Fig. 2(a) is {0.8, 0.9, 0.7, 0.85, 0.33, 0.3, 0.6, 0.36}, the set of similarity measurements in Fig. 2(b) is {0.8, 0.9, 0.55, 0.85, 0.33, 0.3, 0.6, 0.36}.

TABLE I
STATISTICS ASSOCIATED WITH THE EXAMPLE SHOWN IN FIG. 2(A)

$\theta = d_i$	0.36	0.6	0.7	0.8
$K(s > d_i, O)$	4	4	3	2
$K(s > d_i, N)$	1	0	0	0
$P(O\|s > \theta)$	4/5	1	1	1
Number of false alarms	1	0	0	0
Number of misses	0	0	1	2
Number of misclassifications	1	0	1	2

TABLE II
STATISTICS ASSOCIATED WITH THE EXAMPLE SHOWN IN FIG. 2(B)

$\theta = d_i$	0.36	0.55	0.6	0.8
$K(s > d_i, O)$	4	3	3	2
$K(s > d_i, N)$	1	1	0	0
$P(O\|s > \theta)$	4/5	3/4	1	1
Number of false alarms	1	1	0	0
Number of misses	0	1	1	2
Number of misclassifications	1	2	1	2

$\theta)_{\theta=0.55}$. But $\theta = 0.55$ will not be selected, because it produces a larger number of misses. False negative (i.e., missing) is much more harmful than false positive (i.e., false alarm).

VI. FUSION ALGORITHMS

One important question is how to fuse the classification results of different classifiers in a principled manner. Since the cues from textual content and visual content are relatively independent, we are incapable of fusing prior knowledge from one classifier to another classifier like what has been done in [28]. In this paper, we introduce two fusion algorithms: weighting approach, which is manual-based, and Bayesian approach, which is intelligence-based.

A. Weighting Approach

Based on collections of similarity measurements from both text classifier and image classifier, it is straightforward to use a weight to combine the similarities into a similarity measurement as a whole. Let $S_{i,T}$ denote the probability that the ith web page belongs to the phishing category associated with the text classifier, and $S_{i,V}$ denote the similarity of the ith web page and the protected web page. The hybrid similarity measurement is defined by

$$S_{i,W} = \beta \cdot S_{i,T} + (1 - \beta) \cdot S_{i,V} \tag{23}$$

where $\beta \in [0, 1]$ is a weighting parameter that is used to balance the weights of similarity measurements from text and image classifier. We then compare the hybrid similarity measurement $S_{i,W}$ to a predefined threshold θ_W, which also can be statistically estimated by using our Bayesian model reported in Section V. If the similarity measurement $S_{i,W}$ exceeds the threshold θ_W, the web page is classified as phishing, otherwise, the web page is classified as normal.

B. Bayesian Approach

Weighting approach to fusion is straightforward but needs the empirical study on the weighting parameter. In this section we introduce a Bayesian approach that directly fuses the classification results instead of the similarity measurements. Since the similarities $S_{i,T}$ and $S_{i,V}$ are in the range of $[0, 1]$, i.e., $S_{i,T} \in [0, 1]$ and $S_{i,V} \in [0, 1]$, we partition the entire interval $[0, 1]$ into L sub-intervals, i.e., $[I_0, I_1], \ldots, [I_{L-1}, I_L]$. For a given web page, we achieve two classification results by using the text classifier and the image classifier. Let binary random variables $E_T \in \{O, N\}$ and $E_V \in \{O, N\}$ be the events that the web page is phishing or normally decided by the text classifier and the image classifier, respectively. If $E_T = E_V$, i.e., both classifiers make the same decision, we classify the web page into corresponding category. If $E_T \neq E_V$, i.e., the classifiers make different decisions, we obey to the classifier that has a larger probability of correctness conditioning on the distributions of similarity measurements: $S_{i,T} \in [L_{t-1}, L_t]$ $(t = 1, 2, \ldots, L)$ and $S_{i,V} \in [L_{\upsilon-1}, L_\upsilon]$ $(\upsilon = 1, 2, \ldots, L)$. But the important question is how to estimate the posterior probability of correctness associated with a classifier. In this paper, we develop a Bayesian model to handle this issue. Let binary random variable $E_F \in \{C, I\}$ be the event that the classification result of a classifier is correct or incorrect. The desired Bayesian model to determine a posterior probability conditioning on a sub-interval $l_k = [L_{k-1}, L_k]$ for a classifier is given by

$$P(C|l_k) = \frac{P(C)P(l_k|C)}{P(l_k)}. \tag{24}$$

Since

$$P(l_k) = P(C)P(l_k|C) + P(I)P(l_k|I) \tag{25}$$

we obtain

$$P(C|l_k) = \frac{P(C)P(l_k|C)}{P(C)P(l_k|C) + P(I)P(l_k|I)}. \tag{26}$$

The estimated posterior probabilities conditioning on different sub-intervals are used for fusion algorithm to make final decisions on the results. Compared with the traditional fusion strategies [37], [38] such as Dempster–Shafer's theory and classifier combination rules (e.g., sum rule, median rule, etc.), our Bayesian approach builds a bridge between the similarity measurements from different types of features and the single decision of each classifier in an automatic way by modeling the classification experience from different classifiers. We clarify two points here. 1) Different classifiers enable the adoption of different number of sub-intervals in the partitioning process with the same number of known web pages. 2) The number of sub-intervals can be empirically determined in a large set of known data samples.

C. Implementation

According to (26), for a given web page T, a posterior probability conditioning on a sub-interval $l_t = [L_{t-1}, L_t]$ for the text classifier is given by

$$P_T(C|l_t) = \frac{P_T(C)P_T(l_t|C)}{P_T(C)P_T(l_t|C) + P_T(I)P_T(l_t|I)}. \tag{27}$$

For a large set of known web pages, we estimate the posterior probability $P_T(C|l_t)$ by calculating

$$P_T(l_t|C) = \frac{K_T(l_t, C)}{K_T(C)} \tag{28}$$

$$P_T(l_t|I) = \frac{K_T(l_t, I)}{K_T(I)} \tag{29}$$

$$P_T(C) = \frac{K_T(C)}{K_F} \tag{30}$$

$$P_T(I) = \frac{K_T(I)}{K_F} \tag{31}$$

where $K_T(l_t, C)$ and $K_T(l_t, I)$ denote the numbers of correctly classified and incorrectly classified web pages associating their similarity measurements belonging to the sub-interval l_t, respectively, $K_T(C)$ and $K_T(I)$ denote the number of correctly classified and incorrectly classified web pages, respectively, based on the trained text classifier, and $K_F = K_T(C) + K_T(I)$ denotes the total number of web pages in the training set. Substituting (28)–(31) into (27), we obtain

$$P_T(C|l_t) = \frac{K_T(l_t, C)}{K_T(l_t, C) + K_T(l_t, I)}. \tag{32}$$

Likewise, we determine the posterior probability $P_V(C|l_v)$ conditioning on a sub-interval $l_v = [L_{v-1}, L_v]$ for the image classifier by

$$P_V(C|l_v) = \frac{K_V(l_v, C)}{K_V(l_v, C) + K_V(l_v, I)} \tag{33}$$

where $K_V(l_v, C)$ and $K_V(l_v, I)$ denote the numbers of correctly classified and incorrectly classified web pages associating their similarity measurements belonging to the sub-interval l_v, respectively. For a new testing web page, we first locate the corresponding intervals according to its similarity measurements, i.e., determine the index values of t and v, and then calculate a decision factor δ, which is the ratio of the two posterior probabilities in (32) and (33) and described by

$$\delta = \frac{P_T(C|l_t)}{P_V(C|l_v)}. \tag{34}$$

If $\delta \geq 1$, we obey to the decision given by the text classifier rather than the visual classifier, and vice versa in the case of $\delta < 1$. The overall implementation procedures of fusion algorithm are summarized as follows.

Step 1: Input the training set, train a text classifier and an image classifier, and then collect similarity measurements from different classifiers.

Step 2: Partition the interval of similarity measurements into sub-intervals.

Step 3: Estimate the posterior probabilities conditioning on all the sub-intervals for the text classifier according to (32).

Step 4: Estimate the posterior probabilities conditioning on all the sub-intervals for the image classifier according to (33).

Step 5: For a new testing web page, classify it into corresponding category by using the text classifier and the image classifier. If it is classified into different categories, locate the sub-interval that the similarity measurement of the web page belongs to and execute step 6), if else, execute step 7).

Step 6: Calculate the decision factor for the testing web page according to (34).

Step 7: Return the final classification results to a user or a web browser.

VII. Experiments

We conduct a large-scale experiment to evaluate the performances of the text classifier, the image classifier, and the overall framework we have proposed. Using 26 keywords as queries, 10 272 homepage URLs were retrieved from Google: 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. These homepage URLs have been used as normal web pages in [3]. In this paper, the set of protected web page includes eight real web pages, the URLs of which are listed in Table III. We collected a large number of phishing web pages from real phishing attack cases, which were newly reported by PhishTank[5]. Moreover, we filtered out the empty or duplicated web pages in the normal category and the phishing category. Thus, the entire dataset consists of eight sub-datasets corresponding to the real web pages. The web page distribution of the phishing and normal categories for different sub-datasets used in this paper is summarized in Table III. The entire dataset can be downloaded at *www.ee.cityu.edu.hk/~twschow/Phishing_CityU.rar* for other researchers. We first randomly held out 50% of web pages in each dataset corresponding to the protected web page as training data. The remaining web pages served as testing data. All the experiments were performed on a PC with Intel(R) Core(TM) i7 CPU 860@ 2.80 GHz and 6.00 GB memory. The feature extraction programs were written in Java programming language, and the classification algorithms were implemented using MATLAB 7.0.1 on the Windows XP platform.

We evaluate the performance of different classifiers based on five criteria: correct classification ratio (CCR), the calculation of which is given by the ratio of number of correctly classified web pages and total number of web pages, F-score, a weighted average of the precision and recall where the score reaches its best value at 1 and worst value at 0 [39], the Matthews correlation coefficient (MCC), a balanced measure that describes the confusion matrix of true/false positives and negatives-such measure can be used even if the classes are of very different sizes [40], false negative ratio (FNR), the calculation of which is given by the ratio of number of false negatives and number of phishing web pages, and false alarm ratio (FAR), the calculation of which is given by the ratio of number of false alarms and number of normal web pages.

In the following, we first evaluate the performances of the text classifier (Section VI-A) and image classifier (Section VII-B) with the thresholds estimated by Bayesian

[5]Available at http://www.phishtank.com.

TABLE III
WEB PAGE DISTRIBUTION OF CATEGORIES IN SUB-DATASETS

Protected web page	URL	Phishing	Normal	Total
eBay	https://signin.ebay.com	1636	8291	9927
PayPal	https://www.paypal.com/c2	2551	8291	10842
Rapidshare	https://ssl.rapidshare.com/premiumzone.html	489	8291	8780
HSBC	http://www.hsbc.co.uk/1/2/HSBCINTEGRATION/	452	8291	8743
Yahoo	https://login.yahoo.com	204	8291	8495
Alliance-Leicester	https://www.mybank.alliance-leicester.co.uk/index.asp	182	8291	8473
Optus	https://www.optuszoo.com.au/login	101	8291	8392
Steam	https://steamcommunity.com	96	8291	8387

TABLE IV
CLASSIFICATION RESULTS OF TEXT CLASSIFIER WITH DIFFERENT THRESHOLD SETTING STRATEGIES

Protected Web Page	Predefined threshold						Estimated threshold				
	Thr	CCR	F-score	MCC	FNR	FAR	CCR	F-score	MCC	FNR	FAR
eBay	0.20	97.24%	0.9087	0.8977	136/818	1/4145	97.46%	0.9169	0.9060	123/818	3/4145
PayPal	0.25	99.19%	0.9826	0.9774	35/1275	9/4146	98.52%	0.9677	0.9588	76/1275	4/4146
RapidShare	0.10	99.57%	0.9597	0.9581	18/244	1/4146	99.86%	0.9877	0.9869	4/244	2/4146
HSBC	0.10	99.22%	0.9187	0.9180	34/226	0/4145	99.70%	0.9709	0.9694	9/226	4/4145
Yahoo	0.05	98.42%	0.5110	0.5811	67/102	0/4145	99.27%	0.8208	0.8312	31/102	0/4145
Alliance-Leicester	0.05	99.34%	0.8182	0.8293	28/91	0/4145	99.86%	0.9667	0.9660	4/91	2/4145
Optus	0.05	99.57%	0.7805	0.7983	18/50	0/4146	100%	1	1	0/50	0/4146
Steam	0.20	98.86%	0	NaN	48/48	0/4145	99.57%	0.8000	0.7997	12/48	6/4145

approach. We then demonstrate the performance of our overall framework corresponding to different fusion algorithms (Section VII-C). We finally also include the parameter study.

A. Text Classifier

To demonstrate the performance of our proposed probabilistic model for threshold setting, we first compare the results of the naive Bayes text classifier using Bayesian threshold estimation and predefined threshold method, respectively. It is straightforward to set a predefined threshold for the text classifier. In this case, we investigate the results of the threshold varying from 0 to 1 at increments of 0.05 and select the best choice as a comparison. The classification results of different threshold setting strategies are summarized in Table IV, in which "Thr" denotes the best value of predefined thresholds. It is observed that the text classifier using probabilistic model to determine the threshold delivers better performance on the CCR, F-score, MCC, and FNR than the classifier using predefined threshold for all most sub-datasets. The CCR of the classifier with predefined threshold is only slightly higher than that of the classifier with statistically estimated threshold for the "PayPal" sub-dataset, but it delivers larger number of false alarms. For "RapidShare," "HSBC," and "Alliance-Leicester" sub-datasets, we note that the classifier with statistically estimated threshold performs better on the CCR, F-score, MCC, and FNR but at the expense of increasing false alarms. It is also indicated that an appropriate value of predefined threshold in each sub-dataset is relatively small (usually no larger than 0.25). We also investigate the performance of other widely used classifiers such as K-nearest neighbor (KNN) and SVM [41], [42]. We use the MATLABArsenal[6] toolbox to learn a KNN and SVM classifier on the training data. A total comparison of different classifiers is summarized in Table V. It is interesting to observe that KNN and SVM work rather well for the "eBay" and "PayPal" sub-datasets, and SVM outperforms the Bayesian approach designed in this research for all the sub-datasets except for the "Optus" dataset. It is also noted that KNN delivers a consistent superiority in terms of the FNR performance over SVM and Bayesian approach in spite of relatively low CCR. The results from the experiments shown here suggest that, apart from Bayesian classifier, other classifiers may perform well on some datasets with respect to the special measures. But for the sake of statistical description, Bayesian classifier is highly recommended.

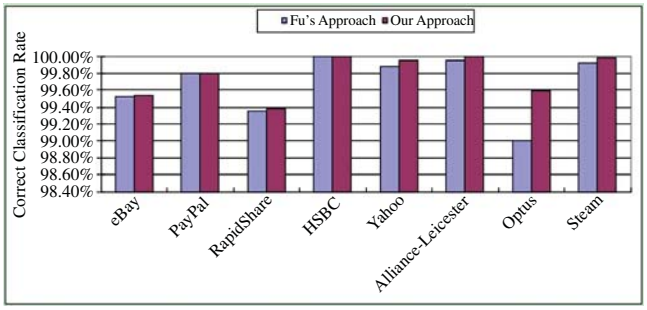

Fig. 3. Comparison of our image classifier and other approaches.

[6]Available at http://www.informedia.cs.cmu.edu/yanrong/MATLABArsenal/MATLABArsenal.zip.

TABLE V
PERFORMANCE OF DIFFERENT TEXT CLASSIFIERS

Protected	KNN			SVM			Bayesian approach		
Web page	CCR	FNR	FAR	CCR	FNR	FAR	CCR	FNR	FAR
eBay	98.73%	10/818	53/4145	99.44%	23/818	5/4145	97.46%	123/818	3/4145
PayPal	99.15%	2/1275	44/4146	99.61%	21/1275	0/4146	98.52%	76/1275	4/4146
RapidShare	98.16%	3/244	78/4146	99.89%	3/244	2/4146	99.86%	4/244	2/4146
HSBC	98.67%	5/226	53/4145	99.84%	6/226	1/4145	99.70%	9/226	4/4145
Yahoo	99.27%	8/102	23/4145	99.69%	13/102	0/4145	99.27%	31/102	0/4145
Alliance-Leicester	97.45%	2/91	106/4145	99.91%	4/91	0/4145	99.86%	4/91	2/4145
Optus	97.81%	1/50	91/4146	99.98%	1/50	0/4146	100%	0/50	0/4146
Steam	96.73%	0/48	137/4145	99.95%	1/48	1/4145	99.57%	12/48	6/4145

TABLE VI
COMPARISON RESULTS OF IMAGE CLASSIFIER USING DIFFERENT APPROACHES

Protected	Fu's approach			Our approach		
Web page	CCR	FNR	FAR	CCR	FNR	FAR
eBay	99.52%	24/818	0/4145	99.54%	23/818	0/4145
PayPal	99.80%	10/1275	1/4146	99.80%	10/1275	1/4146
RapidShare	99.36%	26/244	2/4146	99.38%	26/244	1/4146
HSBC	100%	0/226	0/4145	100%	0/226	0/4145
Yahoo	99.88%	5/102	0/4145	99.95%	2/102	0/4145
Alliance-Leicester	99.95%	2/91	0/4145	100%	0/91	0/4145
Optus	99.00%	0/50	42/4146	99.59%	16/50	1/4146
Steam	99.93%	0/48	3/4145	99.98%	0/48	1/4145

B. Image Classifier

This experiment investigates the performance of the image classifier using the threshold estimated by Bayesian approach. We first compare our method to Fu's approach [3], which uses a dynamic programming method to estimate the threshold and introduces a tolerance parameter to balance false positives and false negatives. We set the parameters in the image classifier to the best optional values recommended by [3]. The comparison results are listed in Table VI and visually illustrated by Fig. 3. It is observed that our method outperforms Fu's approach on both CCR and FAR for "eBay," "RapidShare," "Yahoo," "Alliance-Leicester," "Optus," and "Steam" sub-datasets, and both methods perform equally well for "PayPal" and "HSBC" sub-datasets. We also observe that phishing web pages are completely recognized by our method for "HSBC" and "Alliance-Leicester" sub-datasets. It also shows that our model delivers superior performance on decreasing false negatives for "eBay," "Yahoo" and "Alliance-Leicester" sub-datasets, while Fu's approach shows the significance in the FNR performance for "Optus" dataset. We also compare our probabilistic model to the image classifier using the predefined threshold as shown in Table VII. The predefined threshold varies from 0 to 1 at increments of 0.05. It is observed that our model and the predefined threshold model deliver similar results for most sub-datasets, which indicates that the Bayesian approach to estimate the matching threshold enables us to efficiently capture the statistics of visual similarity measurements.

C. Overall Framework

Weighting-based fusion algorithm described in Section VI-A first combines the similarity measurements from different feature sources into a single similarity framework. This combination is performed by a weight parameter β (23). Then the fusion process by setting a predefined threshold θ_W classifies each testing web page into corresponding category. In Fig. 4, we took "eBay" dataset as an illustrative example, and plotted the classification results of weighting fusion approach by setting a certain pair of parameters (i.e., β and θ_W), which both vary from 0 to 1 at increments of 0.05. From Fig. 4, we observe that there exist pairs of optimal parameters to form a ridge to deliver better classification results. It indicates that these parameters require careful selection by the users.

We compare our Bayesian fusion algorithm to the weighting approach in Table VIII, where the optimal results produced by the weighting approach are summarized as a comparison, and L denotes the number of sub-intervals (Section VI-B). The choice of L depends on different datasets (usually $L = 100$) under our observation from large-scale experiments) and more details on its selection are discussed at the end of this section. Form Table VIII, we observe that our fusion algorithm consistently outperforms the weighting approach based on all the statistical measures (i.e. CCR, F-score, MCC, FNR, and FAR) we include here for all the datasets. For the "PayPal" dataset, more than 35 web pages have been correctly classified by our fusion algorithm compared to the weighting approach.

TABLE VII
CLASSIFICATION RESULTS OF IMAGE CLASSIFIER WITH DIFFERENT THRESHOLD SETTING STRATEGIES

Protected Web page	Predefined threshold						Estimated threshold				
	Thr	CCR	*F*-score	MCC	FNR	FAR	CCR	*F*-score	MCC	FNR	FAR
eBay	0.55	99.50%	0.9845	0.9816	25/818	0/4145	99.54%	0.9857	0.9831	23/818	0/4145
PayPal	0.50	99.80%	0.9957	0.9944	10/1275	1/4146	99.80%	0.9957	0.9944	10/1275	1/4146
RapidShare	0.55	99.41%	0.9437	0.9423	26/244	0/4146	99.38%	0.9417	0.9400	26/244	1/4146
HSBC	0.50	100%	1	1	0/226	0/4145	100%	1	1	0/226	0/4145
Yahoo	0.50	99.95%	0.9901	0.9899	2/102	0/4145	99.95%	0.9901	0.9899	2/102	0/4145
Alliance-Leicester	0.55	100%	1	1	0/91	0/4145	100%	1	1	0/91	0/4145
Optus	0.55	99.38%	0.6487	0.6907	26/50	0/4146	99.59%	0.8000	0.8110	16/50	1/4146
Steam	0.50	99.98%	0.9897	0.9896	0/48	1/4145	99.98%	0.9897	0.9896	0/48	1/4145

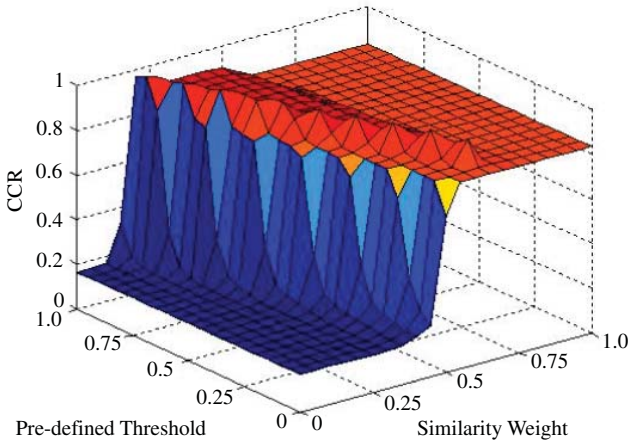

Fig. 4. Similarity weight against predefined threshold associated with weighting approach for "eBay" dataset.

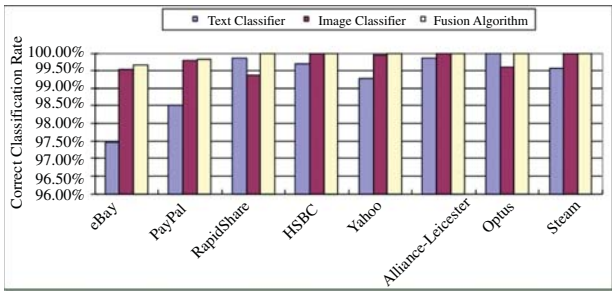

Fig. 5. Overall performances of our proposed schemes.

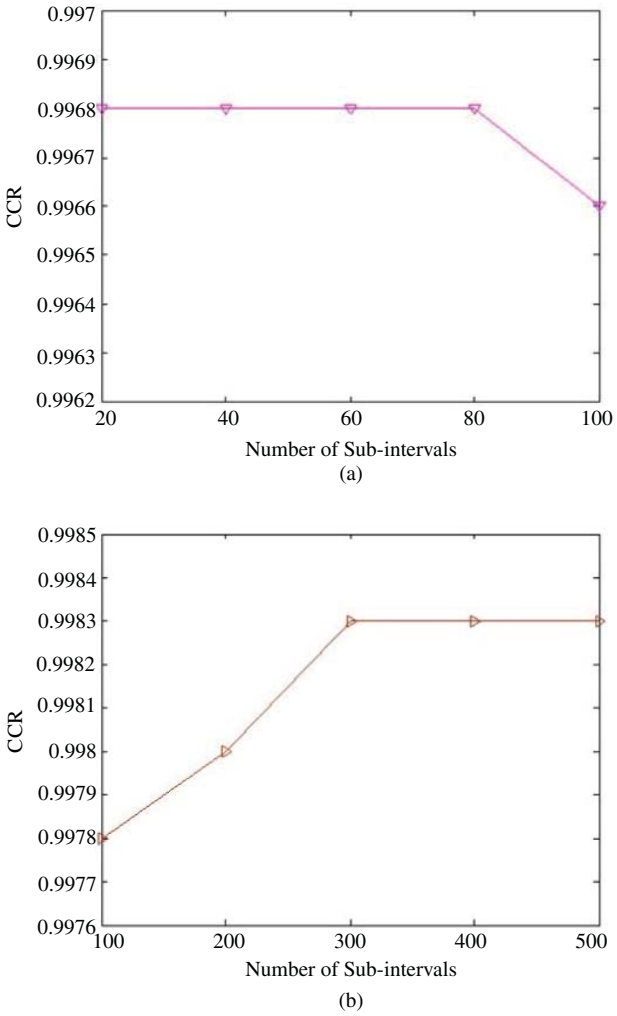

Fig. 6. Number of sub-intervals against CCR. (a) eBay. (b) PayPal.

The Bayesian fusion approach decreases the false alarms and false negatives in a significant rate for "Yahoo" dataset.

Furthermore, to demonstrate the performance of our proposed Bayesian fusion algorithm, we compare it to other classifier combination schemes, i.e., sum rule and median rule, which are the two best classifier combination rules experimentally reported in [38]. The comparative results are summarized in Table IX. It is observed that our proposed Bayesian approach demonstrates superior performance over sum rule and median rule for "RapidShare," "HSBC," "AllianceLeicester," "Optus," and "Steam" datasets, while sum rule delivers the best performance for "Yahoo" datasets in terms of CCR, FNR, and FAR measures. It appears from the results shown in Table IX that the median rule to decrease the FAR does better than the sum rule but at the expense of increasing the FNR.

To get an overview of the total performance of our framework, we summarize the results produced by different classification schemes including the text classifier, the image classifier, and the Bayesian fusion algorithm, which are shown in Fig. 5. It is observed that our fusion algorithm is capable of fusing the results from different classifiers in an efficient manner. Compared to use single classifier, i.e., either the text

TABLE VIII
CLASSIFICATION RESULTS OF OUR FRAMEWORK WITH DIFFERENT FUSION ALGORITHMS

Protected	Weighting approach							Bayesian approach				
Web page	β	θ_W	CCR	F-score	MCC	FNR	FAR	CCR	F-score	MCC	FNR	FAR
eBay	0.95	0.20	99.19%	0.9750	0.9705	37/818	3/4145	99.68%	0.9901	0.9883	16/818	0/4145
PayPal	1.00	0.25	99.19%	0.9826	0.9774	35/1275	9/4146	99.83%	0.9965	0.9954	8/1275	1/4146
RapidShare	0.95	0.10	99.84%	0.9855	0.9847	6/244	1/4146	99.98%	0.9980	0.9978	0/244	1/4146
HSBC	0.85	0.15	99.54%	0.9537	0.9524	20/226	0/4145	100%	1	1	0/226	0/4145
Yahoo	0.80	0.15	98.78%	0.7759	0.7757	12/102	40/4145	99.98%	0.9951	0.9950	1/102	0/4145
Alliance-Leicester	0.90	0.10	99.69%	0.9305	0.9293	4/91	9/4145	100%	1	1	0/91	0/4145
Optus	0.95	0.05	99.71%	0.8636	0.8705	12/50	0/4146	100%	1	1	0/50	0/4146
Steam	0.95	0.05	99.64%	0.8276	0.8304	12/48	3/4145	99.98%	0.9897	0.9896	0/48	1/4145

TABLE IX
PERFORMANCE OF DIFFERENT FUSION APPROACHES

Protected	Sum rule			Median rule			Our approach		
Web page	CCR	FNR	FAR	CCR	FNR	FAR	CCR	FNR	FAR
eBay	99.70%	12/818	3/4145	99.70%	12/818	3/4145	99.68%	16/818	0/4145
PayPal	99.87%	2/1275	5/4146	99.87%	2/1275	5/4146	99.83%	8/1275	1/4146
RapidShare	99.93%	0/244	3/4146	99.86%	4/244	2/4146	99.98%	0/244	1/4146
HSBC	99.91%	0/226	4/4145	99.79%	9/226	0/4145	100%	0/226	0/4145
Yahoo	100%	0/102	0/4145	99.22%	33/102	0/4145	99.98%	1/102	0/4145
Alliance-Leicester	99.95%	0/91	2/4145	99.91%	4/91	0/4145	100%	0/91	0/4145
Optus	99.98%	0/50	1/4146	99.62%	16/50	0/4146	100%	0/50	0/4146
Steam	99.83%	0/48	7/4145	99.71%	12/48	0/4145	99.98%	0/48	1/4145

classifier or the image classifier, the CCR has been improved by using our fusion algorithm, and significant improvement in terms of other evaluation measures has also been achieved as shown in Table VIII.

We conduct an empirical study on the parameter L (the number of sub-intervals) involved in the Bayesian fusion algorithm and show its effect on the results. In Fig. 6, we plotted the results of the changes of L against the correct classification ratio for two datasets: "eBay" and "PayPal." It is observed that the results are not sensitive to the variations of L, but the scale of L depends on the dataset. Under the observations from our large-scale experiments, we usually set the parameter L to 100.

VIII. CONCLUSION

A new content-based anti-phishing system has been thoroughly developed. In this paper, we presented a new framework to solve the anti-phishing problem. The new features of this framework can be represented by a text classifier, an image classifier, and a fusion algorithm. Based on the textual content, the text classifier is able to classify a given web page into corresponding categories as phishing or normal. This text classifier was modeled by naive Bayes rule. Based on the visual content, the image classifier, which relies on EMD, is able to calculate the visual similarity between the given web page and the protected web page efficiently [3]. The matching threshold used in both text classifier and image classifier is effectively estimated by using a probabilistic model derived from the Bayesian theory. A novel data fusion model using the Bayesian theory was developed and the corresponding fusion algorithm presented. This data fusion framework enables us to directly incorporate the multiple results produced by different classifiers. This fusion method provides insights for other data fusion applications. Large-scale experiments were conducted in this paper. Our results corroborated the effectiveness of our proposed framework. Experimental results also suggested that our proposed model is capable of improving the accuracy of phishing detection. More importantly, it is worth noting that our content-based model can be easily embedded into current industrial anti-phishing systems. Despite the promising results presented in this paper, our future work will include adding more features into the content representations into our current model, and investigating incremental learning models to solve the knowledge updating problem in current probabilistic model.

APPENDIX

Let the binary state random variable $Z = \{M, U\}$ represent the event that a web page is mistakenly or correctly classified. The problem of directly minimizing the number of misclassified web pages conditioning on a threshold θ can be formulated to minimize a posterior probability

$$P(M|\theta) = P(O|s \leq \theta) + P(N|s > \theta). \tag{35}$$

Since

$$P(N|(s > \theta)) = P(s > \theta) - P(O|(s > \theta)) \tag{36}$$

and

$$P(O|(s \leq \theta)) = \frac{P(O) - P(O|(s > \theta))P(s > \theta)}{1 - P(s > \theta)} \tag{37}$$

by substituting (36) and (37) into (35), we obtain

$$P(M|\theta) = P(s > \theta) + \frac{P(O) - P(O|(s > \theta))}{1 - P(s > \theta)}. \quad (38)$$

According to (38), if a threshold θ increases, the probability of the event $\{s > \theta\}$ $P(s > \theta)$ will decrease, whilst the probability $P(O|(s > \theta))$ will increase, thus the second item in (38) will decrease accordingly, which makes the probability $P(M|\theta)$ decrease. (Here, the value of $P(M|\theta)$ must belong to the range of [0, 1] when we change the value of θ.) Therefore, maximizing a posterior probability $P(O|s > \theta)$ conditioning on a threshold θ is equal to minimize the number of misclassified web pages.

ACKNOWLEDGMENT

The authors would like to thank the reviewers for their detailed and useful comments.

REFERENCES

[1] A. Emigh. (2005, Oct.). *Online Identity Theft: Phishing Technology, Chokepoints and Countermeasures*. Radix Laboratories Inc., Eau Claire, WI [Online]. Available: http://www.antiphishing.org/phisging-dhs-report.pdf

[2] L. James, *Phishing Exposed.* Rockland, MA: Syngress, 2005.

[3] A. Y. Fu, W. Liu, and X. Deng, "Detecting phishing web pages with visual similarity assessment based on earth mover's distance (EMD)," *IEEE Trans. Depend. Secure Comput.*, vol. 3, no. 4, pp. 301–311, Oct.–Dec. 2006.

[4] *Global Phishing Survey: Domain Name Use and Trends in 1H2009*. Anti-Phishing Working Group, Cambridge, MA [Online]. Available: http://www.antiphishing.org

[5] N. Chou, R. Ledesma, Y. Teraguchi, and D. Boneh, "Client-side defense against web-based identity theft," in *Proc. 11th Annu. Netw. Distribut. Syst. Secur. Symp.*, San Diego, CA, Feb. 2005, pp. 119–128.

[6] M. Wu, R. C. Miller, and S. L. Garfinkel, "Do security toolbars actually prevent phishing attacks?" in *Proc. SIGCHI Conf. Human Factors Comput. Syst.*, Montreal, QC, Canada, Apr. 2006, pp. 601–610.

[7] Y. Zhang, S. Egelman, L. Cranor, and J. Hong, "Phinding phish: Evaluating anti-phishing tools," in *Proc. 14th Annu. Netw. Distribut. Syst. Secur. Symp.*, San Diego, CA, Feb. 2007, pp. 1–16.

[8] L. Li and M. Helenius, "Usability evaluation of anti-phishing toolbars," *J. Comput. Virol.*, vol. 3, no. 2, pp. 163–184, 2007.

[9] M. Aburrous, M. Hossain, F. Thabatah, and K. Dahal, "Intelligent phishing website detection system using fuzzy techniques," in *Proc. 3rd Int. Conf. Inf. Commun. Technol.*, Damascus, VA, Apr. 2008, pp. 1–6.

[10] R. Dhamija and J. D. Tygar, "The battle against phishing: Dynamic security skins," in *Proc. Symp. Usable Privacy Secur.*, Pittsburgh, PA, Jul. 2005, pp. 77–88.

[11] M. Wu, R. C. Miller, and G. Little, "Web wallet: Preventing phishing attacks by revealing user intentions," in *Proc. 2nd Symp. Usable Privacy Secur.*, Pittsburgh, PA, Jul. 2006, pp. 102–113.

[12] E. Gabber, P. B. Gibbons, Y. Matias, and A. J. Mayer, "How to make personalized web browsing simple, secure, and anonymous," in *Proc. 1st Int. Conf. Finan. Cryptograp.*, Anguilla, British West Indies, Feb. 1997, pp. 17–32.

[13] J. A. Halderman, B. Waters, and E. W. Felten, "A convenient method for securely managing passwords," in *Proc. 14th Int. Conf. World Wide Web*, Chiba, Japan, May 2005, pp. 471–479.

[14] W. Liu, N. Fang, X. Quan, B. Qiu, and G. Liu, "Discovering phishing target based on semantic link network," *Future Generat. Comput. Syst.*, vol. 26, no. 3, pp. 381–388, Mar. 2010.

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

[16] T. A. Phelps and R. Wilensky, "Robust hyperlinks and locations," *D-Lib Mag.*, vol. 6, nos. 7–8, pp. 7–8, Jul–Aug. 2000.

[17] P. Likarish, E. Jung, D. Dunbar, T. E. Hansen, and J. P. Hourcade, "B-APT: Bayesian anti-phishing toolbar," in *Proc. IEEE Int. Conf. Commun.*, Beijing, China, May 2008, pp. 1745–1749.

[18] W. Liu, X. Deng, G. Huang, and A. Y. Fu, "An antiphishing strategy based on visual similarity assessment," *IEEE Internet Comput.*, vol. 10, no. 2, pp. 58–65, Mar.–Apr. 2006.

[19] W. Liu, G. Huang, X. Liu, M. Zhang, and X. Deng, "Detection of phishing web pages based on visual similarity," in *Proc. 14th Int. Conf. World Wide Web*, Chiba, Japan, May 2005, pp. 1060–1061.

[20] W. Liu, G. Huang, X. Liu, M. Zhang, and X. Deng, "Phishing web page detection," in *Proc. 8th Int. Conf. Documents Anal. Recognit.*, Seoul, Korea, Aug. 2005, pp. 560–564.

[21] V. Apparao, S. Byrne, M. Champion, S. Isaacs, I. Jacobs, A. Le Hors, G. Nicol, J. Robie, R. Sutor, C. Wilson, and L. Wood. (1998, Oct.). *Document Object Model Level 1 Specification* [Online]. Available: http://www.w3.org/TR/1998/RECDOM-Level-1-19981001

[22] Y. Rubner, C. Tomasi, and L. J. Guibas, "The earth mover's distance as a metric for image retrieval," *Int. J. Comput. Vis.*, vol. 40, no. 2, pp. 99–121, 2000.

[23] M. Chandrasekaran, K. Narayanan, and S. Upadhyaya, "Phishing email detection based on structural properties," in *Proc. 9th Annu. NYS Cyber Secur. Conf.*, New York, Jun. 2006, pp. 2–8.

[24] I. Fette, N. Sadeh, and A. Tomasic, "Learning to detect phishing emails," in *Proc. 16th Int. Conf. World Wide Web*, Banff, AB, Canada, May 2007, pp. 649–656.

[25] S. Abu-Nimeh, D. Nappa, X. Wang, and S. Nair, "A comparison of machine learning techniques for phishing detection," in *Proc. Anti-Phish. Work. Groups 2nd Annu. eCrime Res. Summit*, Pittsburgh, PA, Oct. 2007, pp. 60–69.

[26] R. Basnet, S. Mukkamala, and A. H. Sung, "Detection of phishing attacks: A machine learning approach," in *Soft Computing Applications in Industry*, P. Bhanu, Eds. Berlin, Germany: Springer-Verlag, 2008.

[27] A. McCallum and K. Nigam, "A comparison of event models for naive Bayes text classification," in *Proc. AAAI Workshop Learn. Text Categor.*, Madison, WI, Jul. 1998, pp. 41–48.

[28] W. Hu, O. Wu, Z. Chen, and S. Maybank, "Recognition of pornographic web pages by classifying texts and images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 1019–1034, Jun. 2007.

[29] S. Brin and L. Page, "The anatomy of a large-scale hypertexual web search engine," in *Proc. 7th Int. Conf. World Wide Web*, Brisbane, QLD, Australia, Apr. 1998, pp. 107–117.

[30] M. F. Porter, "An algorithm for suffix stripping," *Program*, vol. 14, no. 3, pp. 130–137, 1980.

[31] C. R. John, *The Image Processing Handbook.* Boca Raton, FL: CRC Press, 1995.

[32] F. Nah, "A study on tolerable waiting time: How long are web users willing to wait?" in *Proc. 9th Amer. Conf. Inf. Syst.*, Tampa, FL, Aug. 2003, p. 285.

[33] T. S. Chua, K. L. Tan, and B. C. Ooi, "Fast signature-based color-spatial image retrieval," in *Proc. IEEE Int. Conf. Multimedia Comput. Syst.*, Ottawa, ON, Canada, Jun. 1997, pp. 362–369.

[34] T. W. S. Chow, M. K. M. Rahman, and S. Wu, "Content based image retrieval by using tree-structured regional features," *Neurocomputing*, vol. 70, nos. 4–6, pp. 1040–1050, 2007.

[35] Y. Liu, Y. Liu, and K. C. C. Chan, "Tensor distance based multilinear locality-preserved maximum information embedding," *IEEE Trans. Neural Netw.*, vol. 21, no. 11, pp. 1848–1854, Nov. 2010.

[36] D. M. Gavrila, "A Bayesian, exemplar-based approach to hierarchical shape matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 8, pp. 1–14, Aug. 2007.

[37] M. Lalmas, "Dempster–Shafer's theory of evidence applied to structured documents: Modeling uncertainty," in *Proc. 20th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, Philadelphia, PA, Jul. 1997, pp. 110–118.

[38] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas, "On combining classifiers," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 3, pp. 226–239, Mar. 1998.

[39] S. M. Beitzel, "On understanding and classifying web queries," Ph.D. thesis, Dept. Comput. Sci., Illinois Institute Technology, Chicago, 2006.

[40] P. Baldi, S. Brunak, Y. Chauvin, C. A. F. Andersen, and H. Nielsen, "Assessing the accuracy of prediction algorithms for classification: An overview," *Bioinformatics*, vol. 16, no. 5, pp. 412–424, 2000.

[41] J. Sun, C. Zheng, X. Li, and Y. Zhou, "Analysis of the distance between two classes for tuning SVM hyperparameters," *IEEE Trans. Neural Netw.*, vol. 21, no. 2, pp. 305–318, Feb. 2010.

[42] F. Angiulli and A. Astorino, "Scaling up support vector machines using nearest neighbor condensation," *IEEE Trans. Neural Netw.*, vol. 21, no. 2, pp. 351–357, Feb. 2010.

Haijun Zhang received the B.Eng. degree in civil engineering and the Masters degree in control theory and engineering from Northeastern University, Shenyang, China, in 2004 and 2007, respectively, and the Ph.D. degree in electronic engineering from the City University of Hong Kong, Kowloon, Hong Kong, in 2010.

He is currently a Post-Doctoral Fellow in the Department of Electrical and Computer Engineering, University of Windsor, ON, Canada. His current research interests include multimedia data mining, machine learning, pattern recognition, evolutionary computing, and communication networks.

Gang Liu received the B.E. degree in computer science from Tsinghua University, Beijing, China. He is currently pursuing the Ph.D. degree in the Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong.

His current research interests include artificial intelligence approaches to computer security and privacy, web document analysis, information retrieval, and natural language processing.

Tommy W. S. Chow (M'94–SM'03) received the B.Sc. (First Hons.) and Ph.D. degrees from the Department of Electrical and Electronic Engineering, University of Sunderland, Sunderland, U.K.

He is currently a Professor in the Department of Electronic Engineering, City University of Hong Kong, Kowloon, Hong Kong. He has been with several consultancy projects including those with the Mass Transit Railway, Kowloon–Canton Railway Corporation, Hong Kong. He has also participated in other collaborative projects with the Hong Kong Electric Company Ltd., the MTR Hong Kong, and Observatory Hong Kong on the application of neural networks for machine fault detection and forecasting. He has authored or co-authored over 130 research papers, contributed five book chapters, and written one book. His current research interests include neural networks, machine learning, pattern recognition, and document analysis and recognition.

Dr. Chow was the recipient of the Best Paper Award from the IEEE Industrial Electronics Society Annual meeting held at Seville, Spain, in 2002.

Wenyin Liu (M'99–SM'02) received the B.Eng. and M.Eng. degrees in computer science from Tsinghua University, Beijing, China, and the D.Sc. degree from Technion, Israel Institute of Technology, Haifa, Israel.

He was a full-time Researcher at Microsoft Research China, Beijing, China. He is currently an Assistant Professor in the Computer Science Department, City University of Hong Kong, Kowloon, Hong Kong. His current research interests include anti-phishing, question answering, graphics recognition, and performance evaluation.

Dr. Liu was awarded the Outstanding Young Researcher Award at the International Conference on Document Analysis and Recognition by the International Association for Pattern Recognition (IAPR) in 2003. He had been TC10 chair of the IAPR for 2006–2010. He is on the Editorial Board of the International Journal of Document Analysis and Recognition. He is a fellow of the IAPR.