MIS Quarterly

SPECIAL ISSUE

DETECTING FAKE WEBSITES: THE CONTRIBUTION OF STATISTICAL LEARNING THEORY[1]

By: **Ahmed Abbasi**
Sheldon B. Lubar School of Business
University of Wisconsin–Milwaukee
Milwaukee, WI 53201
U.S.A.
abbasi@uwm.edu

Zhu Zhang
Department of Management Information Systems
Eller College of Management
University of Arizona
Tucson, AZ 85721
U.S.A.
zhuzhang@u.arizona.edu

David Zimbra
Department of Management Information Systems
Eller College of Management
University of Arizona
Tucson, AZ 85721
U.S.A.
zimbra@u.arizona.edu

Hsinchun Chen
Department of Management Information Systems
Eller College of Management
University of Arizona
Tucson, AZ 85721
U.S.A.
hchen@eller.arizona.edu

Jay F. Nunamaker, Jr.
Department of Management Information Systems
Eller College of Management
University of Arizona
Tucson, AZ 85721
U.S.A.
nunamaker@eller.arizona.edu

Abstract

Fake websites have become increasingly pervasive, generating billions of dollars in fraudulent revenue at the expense of unsuspecting Internet users. The design and appearance of these websites makes it difficult for users to manually identify them as fake. Automated detection systems have emerged as a mechanism for combating fake websites, however most are fairly simplistic in terms of their fraud cues and detection methods employed. Consequently, existing systems are susceptible to the myriad of obfuscation tactics used by fraudsters, resulting in highly ineffective fake website detection performance. In light of these deficiencies, we propose the development of a new class of fake website detection systems that are based on statistical learning theory (SLT). Using a design science approach, a prototype system was developed to demonstrate the potential utility of this class of systems. We conducted a series of experiments, comparing the proposed system against several existing fake website detection systems on a test bed encompassing 900 websites. The results indicate that systems grounded in SLT can more accurately detect various categories of fake websites by utilizing richer sets of fraud cues in combination with problem-specific knowledge. Given the hefty cost exacted by fake websites, the results have important implications for e-commerce and online security.

[1]Raghu Santanam was the accepting senior editor for this paper. Ravi Bapna served as the associate editor.

Keywords: Fake website detection, Internet fraud, design science, statistical learning theory, information systems development, website classification

Introduction

While computer security attacks have traditionally targeted software vulnerabilities, attacks that leverage the information asymmetry of online settings to exploit human vulnerabilities are on the rise. Fraud and deception are highly prevalent in electronic markets, impacting hundreds of thousands of Internet users (Chua and Wareham 2004; Selis et al. 2001). Fake websites have emerged as a major source of online fraud, accounting for billions of dollars in fraudulent revenue at the expense of unsuspecting Internet users (Zhang et al. 2007). A recent study estimates that fake websites comprise nearly 20 percent of the entire Web (Gyongyi and Garcia-Molina 2005). A random sampling of over 105 million web pages revealed that 70 percent of ".biz" and 35 percent of ".us" domain pages analyzed were fake (Ntoulas et al. 2006). The Anti-Phishing Working Group received reports of over 20,000 unique fake websites in January, 2008, alone. In addition to immediate monetary losses, fake websites have long-term trust-related implications for users that can result in a reluctance to engage in future online transactions (Malhotra et al. 2004; Pavlou and Gefen 2005).

In light of these concerns, numerous systems have been proposed for automatic fake website detection (Li and Helenius 2007). Most are lookup systems that rely solely on blacklists comprised of uniform resource locators (URLs) taken from member-reporting databases maintained by online trading communities. The reliance by these systems on people's reports makes them reactive by nature: by the time fake websites are added to the blacklist, many users have already been exposed to them (Chou et al. 2004). A related group of systems use proactive classification techniques, capable of detecting fake websites independently of user reports. These systems utilize fraud cues: important design elements of fake websites that may serve as indicators of their lack of authenticity. Unfortunately, the fraud cues and classification heuristics employed by existing classifier systems are overly simplistic, making them easy to circumvent (Li and Helenius 2007; Zhang et al. 2007). Additionally, fake website detection is a dynamic problem. Fraudsters constantly employ new strategies and utilize newer, more sophisticated technologies (Dinev 2006). Fake website detection systems have not been able to keep pace with advancements by their counterparts. Consequently, these systems are fairly poor in terms of their ability to detect fake websites, with detection rates below 70 percent in most cases (Zhang et al. 2007). Existing systems are also limited with respect to the types of fake websites they are capable of detecting (Abbasi and Chen 2007). There remains a need for fake website detection systems capable of accurately and proactively detecting various categories of fake websites.

In this paper, we discuss the challenges associated with fake website detection and the shortcomings of existing systems. Following the design science paradigm (Hevner et al. 2004), we then propose the creation of a new class of fake website detection systems that leverage methods based on statistical learning theory (Vapnik 1999a, 1999b). Such systems can incorporate large quantities of fraud cues and domain-specific knowledge for effective detection of different categories of fake websites, without over reliance on prior human reporting. In order to illustrate the usefulness of the proposed class of systems, we develop a fake website detection system, called AZProtect, based on statistical learning theory. Using a series of experiments, the enhanced performance of AZProtect is empirically demonstrated in comparison with several existing systems on a test bed encompassing hundreds of real and fake websites.

Fake Websites

The success of fake websites is attributable to several factors, including their authentic appearance, a lack of user awareness regarding them, and the ability of fraudsters to undermine many existing mechanisms for protecting against them. Website quality plays an important role in users' initial trust beliefs regarding a particular website (Gefen and Straub 2003; Koufaris 2002; Lowry et al. 2008). Factors such as the aesthetic appearance of a website can quickly influence users' intentions to transact with a vendor (Everard and Galletta 2005). Fake websites are often very professional-looking and sophisticated in terms of their design (Levy 2004; MacInnes et al. 2005). Their high-quality appearance makes it difficult for users to identify them as fraudulent (Sullivan 2002). In a controlled experiment involving experienced Internet shoppers (i.e., shoppers who regularly made online purchases), more than 82 percent of the test subjects purchased a laptop from a fake website (Grazioli and Jarvenpaa 2000). In two recent studies, 60 percent and 72 percent of test subjects provided personal information to fake websites, respectively (Jagatic et al. 2007; Wu et al. 2006). Below, we briefly discuss categories of fake websites, existing fake website detection systems, and potential fraud cues for identification.

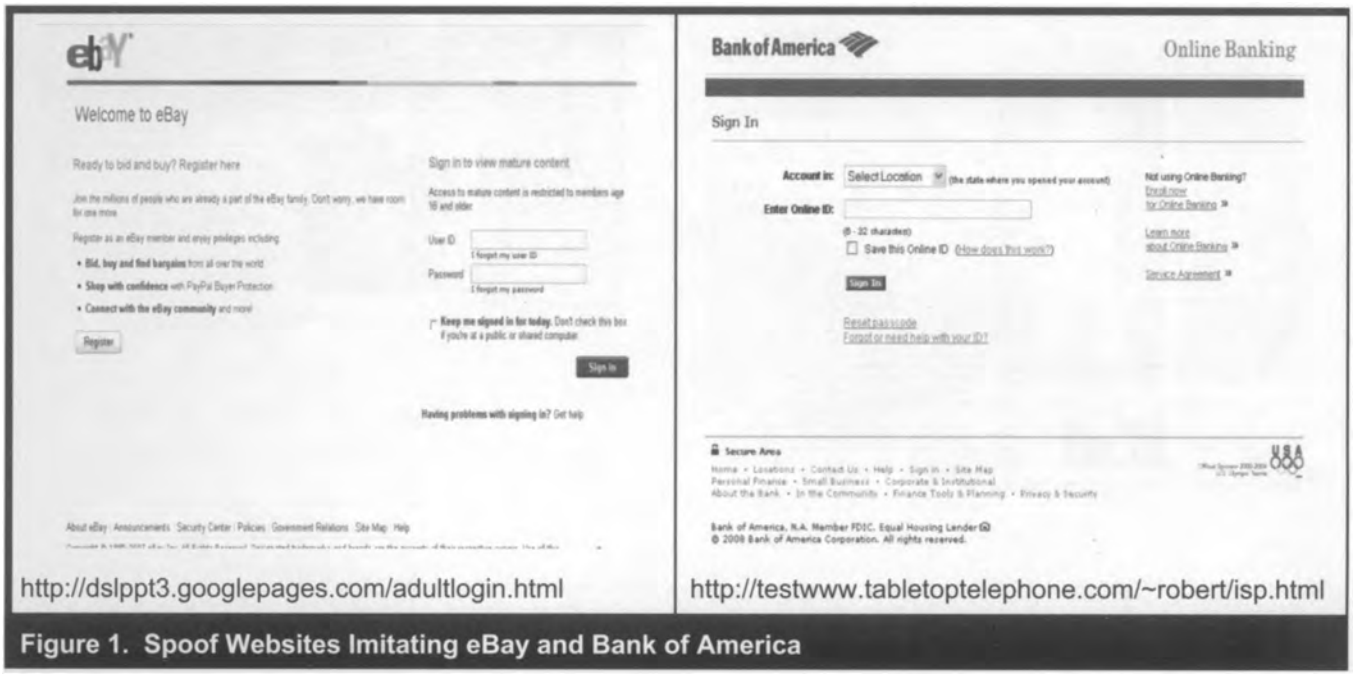


Figure 1. Spoof Websites Imitating eBay and Bank of America

Fake Website Categories

Fake websites fall into two groups: those that target search engines (called web spam) and those that attack web users (Dinev 2006; Gyongyi and Garcia-Molina 2005). We limit our discussion to two categories of the latter, since these have serious implications for e-commerce and Internet fraud (Chua and Wareham 2004): *spoof* and *concocted* sites.

Spoof sites are imitations of existing commercial websites (Chou et al. 2004). Commonly spoofed websites include eBay, PayPal, and various banking and escrow service providers, as depicted in Figure 1. The intention of these sites is online identity theft: deceiving customers of the authentic sites into providing their information to the fraudster operated spoofs (Dinev 2006). Hundreds of new spoof sites are detected daily. These spoofs are used to attack millions of Internet users (Chou et al. 2004; Zhang et al. 2007).

Concocted sites are deceptive websites attempting to appear as unique, legitimate commercial entities (e.g., shipping companies, escrow services, investment banks, online retailers, etc.). The objective of concocted websites is failure-to-ship fraud: taking customers' money without providing the agreed-upon goods or services (Chua and Wareham 2004; Hoar 2005). Concocted sites are also becoming increasingly common, with over 100 new entries added daily to online databases such as the Artists Against 4-1-9 (Airoldi and Malin 2004). Figure 2 shows two concocted investment bank websites.

Fake Website Detection Systems

Fake website detection systems use lookup and/or classification mechanisms for identifying phony websites (Li and Helenius 2007; Wu et al. 2006). They utilize a client-server architecture where the server side maintains a blacklist of known fake URLs (Zhang et al. 2007). The blacklists are taken from online trading communities (Chua et al. 2007), such as the Anti-Phishing Working Group, PhishTank.com, Artists Against 4-1-9, and Escrow-Fraud.com. Many lookup systems also allow users to report websites directly through their interface. The client-side tool checks the blacklist (and in some instances, a user-defined whitelist) and blocks websites that pose a threat. Popular lookup systems include Microsoft's IE Phishing Filter, Mozilla Firefox's Firephish, Sitehound, Cloudmark, and the GeoTrust TrustWatch toolbar (Wu et al. 2006).

Classifier systems detect fake websites based on the appearance of fraud cues in website content and/or domain registration information. Many classifier systems also utilize a blacklist, where the classifier is only applied to URLs not appearing on the blacklist. Existing classifier systems use simple, static, rule-based heuristics and limited fraud cues, making them susceptible to easy exploits (Zhang et al. 2007). SpoofGuard uses image hashes, password encryption checks, URL similarities, and domain registration information (Chou et al. 2004). Netcraft's classifier relies solely on domain registration information: domain name, host name, host country and the registration date of the website (Wu et al. 2006).

Figure 2. Concocted Investment Company Websites

eBay's account guard compares the visited website's content against that of eBay and PayPal's websites. Websites that appear overly similar to eBay or Paypal are blocked (Zhang et al. 2007). Sitewatcher compares the visual and text similarity of the website of interest against a user-defined whitelist (Liu et al. 2006).

Table 1 presents a summary of existing fake website detection systems. For each system, it describes the detection mechanisms employed (i.e., classifier or lookup), the categories of fake websites for which the system is intended, and prior performance results on spoof sites. The results, which were taken from Zhang et al. (2007), consist of the overall results of the system on 716 legitimate and spoof websites, and the detection rates on 200 spoof sites. As revealed in Table 1, the performance of current fake website detection systems is inadequate. Most are lookup systems intended to detect spoof sites; with many having low spoof detection rates. Since few of these systems are designed to handle concocted websites, it is likely that their performance on such fake websites might be even worse.

The poor performance of existing systems impacts their effectiveness; users are distrusting of their recommendations, even when they are correct (Li and Helenius 2007). While conducting a user study on existing systems, Wu et al. (2006; p. 601) observed that users "disregarded or explained away the toolbars' warnings if the content of the web pages looked legitimate." Zhang et al. (2007) noted that current systems' heavy reliance on blacklists has resulted in inadequate fake website detection rates. There is a need for fake website detection systems that utilize a rich set of fraud cues (presented in the following section) coupled with improved classification methods for augmented detection of spoof and concocted websites. Such systems could greatly benefit Internet users that lack the expertise necessary to detect fake websites on their own.

Fake Website Fraud Cues

Fake websites often use automatic content generation techniques to mass produce fake web pages (Urvoy et al. 2006). Although this expedites the fake website development process, it also results in many design similarities which may be discernable by comparing new fake websites against databases of existing fakes (Fetterly et al. 2004). Online trading communities and prior research have identified categories of fraud cues pervasive in fake websites, yet seldom used in existing detection systems (Kolari et al. 2006). These fraud cue categories span all three major components of website design: information, navigation, and visual design (Cyr 2008; Lowry et al. 2008; McKnight et al. 2002). *Web page text* often contains fraud cues stemming from information design elements (Wu and Davidson 2006). The *linkage* information and *URL* names for a website can provide insightful fraud cues relating to navigation design characteristics (Kolari et al. 2006). Fraud cues pertaining to visual design are commonly manifested in a web page's *source code* and *images* (Urvoy et al. 2006). These cue categories are discussed below.

Table 1. Summary of Existing Fake Website Detection Systems

System Type	Tool Name	Fraud Cues Used	Website Type	Prior Results (Spoof Sites)
Look Up	Cloudmark	Server-side blacklist	Spoof sites	Overall: 83.9% Spoof Detection: 45.0%
	EarthLink Toolbar	Server-side blacklist	Spoof sites	Overall: 90.5% Spoof Detection: 68.5%
	FirePhish	Server-side blacklist	Spoof sites	Overall: 89.2% Spoof Detection: 61.5%
	GeoTrust TrustWatch	Server-side blacklist	Spoof sites	Overall: 85.1% Spoof Detection: 46.5%
	IE Phishing Filter	Client-side whitelist and server-side blacklist	Spoof sites	Overall: 92.0% Spoof Detection: 71.5%
Classifier	CallingID	Domain registration information and server-side blacklist	Spoof sites	Overall: 85.9% Spoof Detection: 23.0%
	eBay Account Guard	Text and image content similarity to eBay and Paypal websites and server-side blacklist	Spoof sites (primarily eBay and PayPal)	Overall: 83.2% Spoof Detection: 40.0%
	Netcraft	Domain registration information and server-side blacklist	Concocted sites, spoof sites	Overall: 91.2% Spoof Detection: 68.5%
	SiteWatcher	Text and image feature similarity, stylistic feature correlation, and client-side whitelist	Spoof sites	Not Evaluated
	SpoofGuard	Image hashes, password encryption, URL similarities, domain registration information	Concocted sites, spoof sites	Overall: 67.7% Spoof Detection: 93.5%

Web Page Text

Fraud cues found in a web page's text include misspellings and grammatical mistakes; which are more likely to occur in illegitimate websites (Selis et al. 2001). Other useful text fraud cues are lexical measures (e.g., total words per page, average sentence length), and the frequency of certain word phrases (Ntoulas et al. 2006). Concocted websites also make elaborate use of trust-enabling features such as customer testimonials and "frequently asked questions" sections (Grazioli and Jarvenpaa 2000). However, the text is often similar to prior fake websites.

Web Page Source Code

Web page source code elements, including hypertext markup language (HTML) commands for transferring data and redirecting to other websites, are frequently utilized in fake web pages (Chou et al. 2004; Drost and Scheffer 2005). For spoof sites, javascript embedded in the HTML source code is often used to conceal the websites' true identities from users (Dinev 2006). Design similarities can also be detected from source code, by measuring the occurrence and sequence of the two pages' HTML tags (Urvoy et al. 2006; Wu and Davidson 2006).

URLs

Lengthier URLs, and ones with dashes or digits are common in fake websites, as shown in Figure 1 (Fetterly et al. 2004). URLs using "http" instead of "https" and ones ending with ".org," ".biz," ".us," or ".info" are also more likely to be fake (Drost and Scheffer 2005).

Images

Fake websites frequently use images from existing legitimate or prior fake websites. Spoof sites copy company logos from the websites they are mimicking (Chou et al. 2004). Concocted websites reuse images of employees, products, customers, and company assets from older concocted sites (Abbasi and Chen 2007).

Linkage

Linkage-based fraud cues include the frequency of in and out link information at the site (between websites) and page (between pages in the same site) levels (Abbasi and Chen 2007; Drost and Scheffer 2005). The number of relative (e.g., ../../default.htm) and absolute (e.g. http://www.abc.com/default.htm) address links are also useful cues, since fraudsters often use relative links when mass producing fake websites (Wu and Davidson 2006).

Fraud Cue Challenges

Utilizing extended fraud cue sets, along with appropriate problem-specific domain knowledge pertaining to those fraud cues, is not without its challenges. Fraud cues are inherent in diverse website design elements. A website contains many pages, with each page comprised of multiple images, along with body text, source code, URLs, and structural attributes based on the page's linkage with other web pages. There are representational difficulties associated with integrating a heterogeneous set of website attributes into a classifier system (Tan and Wang 2004; Yu et al. 2004). Furthermore, two important fake website characteristics are stylistic similarities and content duplication across phony websites. Similarly, prior research on linked document classification has noted that certain structural/linkage-based attributes such as the number of in/out links and page levels are at least as important and effective as various page content-based attributes (Drost and Scheffer 2005; Shen et al. 2006; Wu and Davison 2006). Leveraging these key problem-specific elements into a classification system could be highly useful; however, standard classification methods are not suitable for handling such complexities (Kolari et al. 2006; Wu and Davison 2006). Another issue is the dynamic nature of online fraud (Levy 2004). Dinev (2006; p. 81) noted that "substantial technological improvements distinguish recent spoof sites compared to their predecessors from even only a year ago." Fake website detection requires the constant revision of attributes used to represent various fraud cue categories in a manner analogous to the periodic signature updating mechanisms incorporated by e-mail spam blockers and anti-virus software.

Fake Website Detection Using Methods Based on Statistical Learning Theory ■

The deficiencies of existing fake website detection systems, coupled with the challenges associated with building more effective ones, warrants the use of guidelines to help inform future system development. The design science paradigm provides concrete prescriptions for understanding problems related to information systems development (March and Smith 1995; Nunamaker et al. 1991). Information systems design is a product and a process (Hevner et al. 2004). The design product encompasses guidelines for the development of *IT artifacts* (Walls et al. 1992). IT artifacts can be broadly defined as constructs (vocabulary and symbols), models (abstractions and representations), methods (algorithms and practices), and instantiations (implemented and prototype systems). The design process entails building and evaluating those artifacts (March and Smith 1995); it is centered on the notion of devising artifacts through a systematic *search* process, and eventually *evaluating* the utility of the artifacts to attain goals (Hevner et al. 2004; Simon 1996). In our research, the artifacts we intend to deliver are methods and instantiations. We intend to build a prototypical fake website detection system (instantiation) that provides "proof by construction" (Nunamaker et al. 1991). The system will encompass a core algorithm (method) for detecting fake websites.

The generation of suitable guidelines (i.e., design product) for the development of enhanced fake website detection systems presents a key design challenge. Based on our review of fake website types, existing systems, and potential fraud cue categories presented earlier, we have identified four important characteristics that fake website detection systems must possess. They should

(1) Exhibit the ability to generalize across diverse and vast collections of concocted and spoof websites
(2) Incorporate rich sets of fraud cues
(3) Leverage important domain-specific knowledge regarding the unique properties of fake websites: stylistic similarities and content duplication
(4) Provide long-term sustainability against dynamic adversaries by adapting to changes in the properties exhibited by fake websites

When faced with the creation of information systems that support challenging problems lacking sufficient design guidelines, many studies have emphasized the need for design theories to help govern the development process (Markus et al. 2002; Walls et al. 1992). These can be new or existing theories used to facilitate the systematic creation of new IT artifacts (Abbasi and Chen 2008a). Given the lack of prece-

dence for the design and development of effective fake website detection systems, we turn to the machine learning literature.

Machine learning algorithms use an inductive learning process where a classifier is built by observing the characteristics of a set of instances manually categorized into appropriate classes (called training data). For each class in the training data, the inductive process discovers characteristics that a new, unseen instance should have in order to be considered a member of that class (Sebastiani 2002). Many theoretically grounded machine learning algorithms have been proposed. Information theory (Shannon 1948) presented the information entropy heuristic, which is at the core of the ID3 decision tree algorithm (Quinlan 1986). Bayes theorem (Bayes 1958) formed the basis for the naïve Bayes and Bayesian network algorithms. Artificial neural networks, including multi-layer perceptrons and Winnow (Littlestone 1988), are inspired by neurobiology. In recent years, statistical learning theory (Vapnik 1999a) has prompted the development of highly effective algorithms that have outperformed comparison machine learning methods in various application domains (Dumais et al. 1998; Ntoulas et al. 2006; Zheng et al. 2006). Statistical learning theory also provides appropriate guidelines to facilitate the development of systems capable of supporting the four necessary characteristics for fake website detection. An overview of statistical learning theory is provided in the ensuing section, followed by a discussion of how it relates to fake website detection.

Statistical Learning Theory: An Overview

Statistical learning theory (SLT), also known as the Vapnik-Chervonenkis theory, is a computational learning theory that attempts to explain the learning process from a statistical point of view. SLT has four key components (Vapnik 1999a, 1999b):

- Theory of consistency of learning processes, which pertains to the conditions for consistency of a learning process based on the empirical risk minimization principle
- Non-asymptotic theory of the rate of convergence of learning processes
- Theory of controlling the generalization ability of learning processes, which deals with the question of how to control the rate of convergence of the learning process
- Theory of constructing learning machines that can execute the learning process and control the generalization ability

SLT, in particular its last component, has motivated the introduction of a very well-known class of learning algorithms: kernel machines. Support vector machines (SVM) is a prime example of an SLT-based learning algorithm that utilizes kernels (Cristianini and Shawe-Taylor 2000). Kernel machines approach the learning problem by mapping input data into a high dimensional feature space where a variety of methods can be used to find relations in the data. Kernel machines owe their name to the use of kernel functions, which enable them to operate in a feature space without explicitly computing its coordinates. Rather, the kernel function simply computes the similarity between pairs of data points in the feature space (Muller et al. 2001; Vapnik 1999b). This allows SLT-based classification algorithms to incorporate large sets of input attributes in a highly scalable and robust manner, an important characteristic for fake website detection (Cristianini and Shawe-Taylor 2000). The kernel function also plays an integral role in enabling a learning process that retains the original, often semantically rich, representation of data relations (Burges 1998; Muller et al. 2001). Given a number of real and fake websites, the kernel machine would enable the use of a kernel function (equipped with domain-specfic knowledge) to compute the similarity between these websites based on the occurrence values of various fraud cues.

Formally, given an input space $\boldsymbol{X}$, in this case the set of all possible websites to be examined, the learning problem can be formulated as finding a classifier

$$C: \boldsymbol{X} \rightarrow \boldsymbol{Y}$$

where $\boldsymbol{Y}$ is a set of possible labels (in this case "real" or "fake") to be assigned to the data points.

Within the SLT framework, finding C relies on a kernel function K that defines a mapping

$$K: \boldsymbol{X} \times \boldsymbol{X} \rightarrow [0, \infty)$$

from the input space X to a similarity score

$$K(x_i, x_j) = f(x_i) \times f(x_j)$$

where x_i and x_j represent two data points, in this case two websites; $f(x_i)$ is a function that maps $\boldsymbol{X}$ to a higher dimensional space without needing to know its explicit representation. This is often referred to as the "kernel trick" (Cristianini and Shawe-Taylor 2000).

Searching for an optimal C involves evaluating different parameters, where α denotes a specific choice of parameter values for the function $f(x, \alpha)$. These parameters are analo-

gous to the weights and biases incorporated within a trained neural network classifier (Burges 1998). Mathematically, the search for the best C can be formulated as a quadratic programming problem that minimizes the sum of the classifier's error rate on training data and its Vapnik-Chervonenkis dimension, a measure of the capacity of the set of functions to which $f(x, \alpha)$ belongs (i.e., their ability to generalize across testing data instances). This value represents an upper bound on $R(\alpha)$, the expected error rate on testing data for a given C (i.e., a trained machine with a specific choice of parameters α), resulting in a classification model capable of providing a combination of accuracy and generalization ability (Burges 1998). The mathematical formulation is as follows:

$$R(\alpha) \leq \frac{1}{2l}\sum_{i=1}^{l}|y_i - f(x_i,\alpha)| + \sqrt{\frac{h(\log(2l/h)+1)-\log(\eta/4)}{l}}$$

where: l is the number of instances in our training data set
i is a particular training instance
y_i is the class label of instance i where $y_i \in \{-1, 1\}$
α is the parameter values for the selected function $f(x, a)$
h is the VC dimension for the set of functions to which $f(x, a)$ belongs
η is a number on the range $0 \leq \eta \leq 1$ signifying the confidence level

In practice, the search process is governed by the "maximum margin" principle (Cristianini and Shawe-Taylor 2000). Intuitively, a good separation of the space is achieved by the hyperplane that has the largest distance to the neighboring data points of both classes. The hope is that the larger the margin or distance between these parallel hyperplanes, the better the generalization error will be. For fake website detection, this translates into a classifier that could potentially generalize across multiple domains and a large number of sites, with high performance levels.

Detecting Fake Websites Using Statistical Learning Methods

Given the strong theoretical foundation of SLT and the efficiency of SVMs as computational tools, they have been successfully applied in many areas, such as pattern recognition (Chen et al. 2005), data mining (Zhou and Wang 2005), text mining (Sun et al. 2004), and web mining (Yu et al. 2004). SLT also provides a mechanism for addressing the four important characteristics necessary for effective fake website detection systems.

Ability to Generalize

The essence of learning from data is generalization, instead of memorization. In other words, statistical models are built by generalizing patterns in existing data, in anticipation of being applied to unseen data and making predictions. The "maximum margin" principle and the corresponding optimization techniques employed by SLT-based classifiers set out to minimize the classification error of classifiers while simultaneously maximizing their generalization capabilities (Burges 1998; Shawe-Taylor and Cristianini 2004). This makes classifiers such as SVM highly suitable for fake website detection, given the demand for proactive systems capable of classifying previously unseen websites.

Rich Fraud Cues

Previous research has identified a large number of website fraud cues, spanning multiple components of website design, from information content and navigation structure to visual presentation. The set of fraud cues required to represent these design elements for accurate fake website detection may encompass thousands of attributes. One of the strengths of SLT-based classifiers is their ability to handle large and potentially heterogeneous feature sets (Joachims 2002; Yu et al. 2004). In the case of fake website detection, SLT-based classifiers transform input data (fraud cue values for various websites) into a kernel matrix of similarity scores between websites. Therefore, they are able to utilize sizable input feature spaces.

Utilization of Domain Knowledge

By supporting the use of domain specific custom kernels, SLT-based classifiers are able to incorporate unique problem nuances and intricacies, while preserving the semantic structure of the input data space (Burges 1998; Joachims 2002; Tan and Wang 2004). Custom kernels have been devised for many classification problems, including the use of string (Lodhi et al. 2002) and tree kernels (Zelenko et al. 2003) for text categorization. Prior studies have noted that fake website detection could greatly benefit from the use of custom kernel functions, although none have been proposed (Drost and Scheffer 2005; Kolari et al. 2006). Fake websites often exhibit templatic properties attributable to style similarities and content duplication across websites. Encoding this knowledge using flat vectors of variables is difficult, because of the inherently nonlinear structure of websites and the interactions between various website components. However, an appropriate kernel function could be used to represent such information.

Dynamic Learning

A significant challenge associated with fake website detection is the dynamic nature of fake websites, as well as their underlying fraud cues (Dinev 2006). They evolve over time, and not always in a predictable fashion. Given the adversarial nature of fake website detection, the classification models used need constant revision. As with other learning-based classifiers, SLT-based classifiers can also update their models by relearning on newer, more up-to-date training collections of real and fake websites (Cao and Gu 2002).

Research Hypotheses

It is important to note that support for the four aforementioned characteristics is not exclusive to SLT-based learning methods. For example, rule-based classifiers also use domain knowledge (Chou et al. 2004; Russell and Norvig 2003). Similarly, various generalizable non-SLT based algorithms have been proposed (Dumais et al. 1998; Sebastiani 2002; Zheng et al. 2006). However, we are unaware of any other classification method that incorporates all four characteristics in unison. The ability of SLT-based classifiers to combine these characteristics in a synergistic fashion makes them more suitable. Accordingly, we present research hypotheses to test the efficacy of using SLT-based learning methods for fake website detection. The hypotheses employ evaluation metrics incorporated in prior fake website detection research (Fu et al. 2006; Liu et al. 2006). Overall accuracy measures the percentage of all websites (real and fake) that are correctly classified. By considering real and fake websites, this metric takes into account false positives and negatives (Zhang et al. 2007). Class-level recall assesses the detection rates for a particular class of websites—legitimate, concocted, or spoof (Chou et al. 2004). For instance, spoof website recall measures the percentage of all spoof websites that were correctly classified as fake. Class-level precision measures accuracy on the subset of websites classified as belonging to that class (Arazy and Woo 2007). For example, spoof website precision measures the percentage of all websites deemed to be spoofs that are indeed fake. It is worth noting that fake website precision is heavily correlated with legitimate website recall (the same being true for legitimate website precision and fake website recall). Therefore, the hypotheses only utilized fake website precision and recall (i.e., precision and recall on concocted and spoof sites) to avoid redundancy.

$$\text{Overall Accuracy} = \frac{\text{Number of Correctly Classified Instances}}{\text{Total Number of Instances}}$$

$$\text{Class - Level Recall} = \frac{\text{Number of Correctly Classified Class Instances}}{\text{Total Number of Instances in Class}}$$

$$\text{Class - Level Precision} = \frac{\text{Number of Correctly Classified Class Instances}}{\text{Total Number of Instances Classified as Belonging to Class}}$$

Our first intuition is that lookup systems, the family of primitive detection methods, are inherently inadequate (Zhang et al. 2007). Philosophically, a lookup table is not a parsimonious representation of substantial intelligence due to the absence of generalization power. For example, a spoof website A, although very similar to a known item B in the blacklist, will not be identified by a pure lookup system if A has not been registered (Wu et al. 2006). Therefore, it is likely that any nontrivial classifier system, rule or learning-based, will outperform systems relying exclusively on a lookup mechanism in terms of overall accuracy and fake website recall.

H1a: *Classifier systems will outperform lookup systems in terms of overall accuracy.*

H1b: *Classifier systems will outperform lookup systems in terms of fake website recall.*

Since lookup systems employ collaborative sanctioning mechanisms similar to those used for reputation ranking, their underlying blacklists tend to be fairly rigorously constructed (Hariharan et al. 2007; Jøsang et al. 2007). Potential website entries are evaluated by online community members with considerable expertise, resulting in highly precise blacklists with few false negatives (Li and Helenius 2006; Wu et al. 2006). In contrast, classifier systems are more likely to misclassify legitimate websites as fake (Zhang et al. 2007).

H1c: *Classifier systems will underperform lookup systems in terms of fake website precision.*

As implied in H1, a classifier system can be rule-based or learning-based (Russell and Norvig 2003). Both rule-based and learning-based approaches seek parsimonious models of knowledge. Rule-based classifiers rely on a set of manually generated classification heuristics stemming from domain knowledge. Given the breadth of applicable rules, sufficiently encoding all necessary rules is a mundane, expensive, and generally infeasible process. Consequently, existing rule-based fake website detection systems are comprised of relatively small rule sets (e.g., Chou et al. 2004). Furthermore, the encoded rules may not be correct due to over reliance on the coder's observations (Zhang et al. 2007). In contrast, learning-based classifiers are better suited to learn classification models using large numbers of fraud cues and training data (i.e., website instances). Hence, we believe the scalability and adaptability of learning-based website classifiers, in particular SLT-based ones, is more advantageous.

H2a: *SLT-based classifiers will outperform rule-based classifiers in terms of overall accuracy.*

H2b: *SLT-based classifiers will outperform rule-based classifiers in terms of fake website recall.*

H2c: *SLT-based classifiers will outperform rule-based classifiers in terms of fake website precision.*

Within the learning-based paradigm, we believe SLT-motivated learners, specifically SVMs, have an edge compared to other traditional machine learning algorithms. SVM's generalization power, warranted by the utilization of kernel functions and the maximum margin principle, presents an important advantage over other classification methods (Muller et al. 2001). Other learning algorithms, such as naïve Bayes, decision trees, and neural networks, embody bright intuitions, yet suffer from representational limitations (Duda et al. 2000; Joachims 2002). Consequently, the supremacy of SLT-motivated learning methods has been hinted by the benchmarking efforts of several studies across multiple learning algorithms (Meyer et al. 2003). SVM has outperformed comparison algorithms on a bevy of classification tasks, including topic and style categorization of text documents (Dumais et al. 1998; Zheng et al. 2006).

H3a: *SLT-based learning classifiers will outperform other traditional learning algorithms in terms of overall accuracy.*

H3b: *SLT-based learning classifiers will outperform other traditional learning algorithms in terms of fake website recall.*

H3c: *SLT-based learning classifiers will outperform other traditional learning algorithms in terms of fake website precision.*

Within the SLT framework, several generic kernel functions have been proposed, with linear, polynomial, and radial basis function being the most common (Burges 1998; Cristianini and Shawe-Taylor 2000). Linear kernels use straight lines to segment data points in the hyperplane, while polynomial ones use curves and radial basis kernels use elliptical functions (Muller et al. 2001). Although benefitting from various facets of SLT such as the maximum margin principle, these classical kernels make generic assumptions about the problem structure (Burges 1998; Tan and Wang 2004). Whenever possible, domain knowledge should be incorporated into the learning process (Guyon and Elisseeff 2002). Kernel functions provide the ideal opportunity for utilizing domain-specific characteristics in the learning process (Zelenko et al. 2003). SLT-based classifiers could greatly benefit from well-designed custom kernel functions capable of incorporating the integral nuances of fake websites (Drost and Scheffer 2005). Accordingly, we believe that SLT-based classifiers, equipped with custom, problem-specific kernel functions that can better preserve important fraud cue relations, will result in improved fake website detection performance.

H4a: *An SLT-based classifier using a well-designed custom kernel will outperform the ones using generic kernel functions in terms of overall accuracy.*

H4b: *An SLT-based classifier using a well-designed custom kernel will outperform the ones using generic kernel functions in terms of fake website recall.*

H4c: *An SLT-based classifier using a well-designed custom kernel will outperform the ones using generic kernel functions in terms of fake website precision.*

A Fake Website Detection System Based on Statistical Learning

Having devised a set of design guidelines, we turn our attention to the design process. Two important design processes for information systems development are *build* and *evaluate*: the construction of IT artifacts and assessment of their effectiveness (March and Smith 1995; Nunamaker 1992; Simon 1996). In this section, we describe the development of two proposed IT artifacts: an instantiation and a method. In order to assess the efficacy of fake website detection systems grounded in SLT, we developed AZProtect. AZProtect uses an extended set of fraud cues in combination with a support vector machines (SVM) classifier. The classifier uses an embedded custom kernel function that is tailored to detect concocted and spoof websites. The fraud cues and classification method employed by AZProtect are expounded upon below.

Extended Fraud Cue Set

AZProtect utilizes a rich fraud cue set comprised of attributes stemming from the five previously mentioned categories: web page text, source code, URLs, images, and linkage. These fraud cues were derived from a training data set of 1,000 legitimate, spoof, and concocted websites collected over a 3 month period. The collection contained nearly 200,000 web pages and 30,000 image files. Initially, over 600,000 potential fraud cue attributes were extracted. Each of these attributes was weighted using the information gain heuristic, based on their occurrence distribution across the

1,000 websites. Attributes that occurred more frequently in either legitimate or fake websites (as compared to the other category) received higher weights. Consistent with prior research, a threshold was used to determine the number of attributes to include in the fraud cue set, where all attributes weighted higher than the threshold were incorporated (Abbasi et al. 2008; Arazy and Woo 2007). Following best practices, we assessed the performance impact of different thresholds and selected one that resulted in a good balance between classification accuracy and feature set size, with smaller sizes being more desirable for computational reasons (Guyon and Elisseeff 2003; Forman 2003). This resulted in the utilization of approximately 6,000 fraud cues in the AZProtect system. Examples of these fraud cues are presented in Table 2.

The web page text cues encompass over 2,500 word phrases, lexical measures, and spelling and grammatical mistakes. Based on Table 2, it is apparent that fraud cues contained in web page text are more likely to occur in concocted websites, since spoof sites replicate the text from legitimate websites (Dinev 2006). In contrast, the fraud cues from the other four categories are applicable to concocted and spoof websites. The URL fraud cues are 1,500 words and characters derived from the URL and anchor text that are pervasive in concocted and spoof websites. Source code fraud cues include 1,000 items pertaining to code commands as well as general programming style markers, both of which have been shown to be useful in related detection tasks (Abbasi and Chen 2008b; Krsul and Spafford 1997; Urvoy et al. 2006). The image features include pixel color frequencies arranged into 1,000 bins as well as 40 image structure attributes (e.g., image height, width, file extension, file size). These attributes are intended to detect the presence of duplicate images in concocted and spoof sites, that is, ones copied from prior websites (Chou et al. 2004). Linkage-based cues span approximately 50 attributes related to the number of incoming and outgoing links at the site and page levels (Wu and Davison 2006). Based on our analysis of 1,000 real and fake websites, fake sites tend to have less linkage than legitimate ones.

SVM Classifier with Custom Kernel Function

A major strength of the SLT-based SVM classifier is that it allows for the utilization of kernel functions that can incorporate domain-specific knowledge, resulting in enhanced classification capabilities. AZProtect uses an SVM classifier equipped with a kernel function that classifies web pages within a website of interest as real or fake. Classification is performed at the web page level; all pages within a website of interest are independently classified. The aggregate of these page level classification results (i.e., the percentage of pages within a website that are classified as fake) is used to determine whether a website is real or fake.

In order to train the SVM classifier, a kernel matrix encompassing values derived from web pages belonging to the training websites is used. This kernel matrix is generated by applying the composite kernel function to all web pages in the training data set, with each web page compared against all other websites in the training data. Using this kernel matrix as input into the formulation described earlier, the SVM algorithm uses the maximum margin principle to find the best linearly separable hyperplane. For each testing web page, the kernel function is applied in the same manner (by comparing it against the training web sites). Using the function's values as input, the trained SVM classifier assigns each test web page a binary class label: 1 for "legit" and 2 for "fake."A description of the custom linear composite kernel is presented below.

Given a website of interest, the proposed linear composite kernel computes the similarity for each web page a in that site against all web pages belonging to b, where b is part of the set of 1,000 real and fake websites in the training data set. Zelenko et al. (2003, p. 1087) noted that "it is critical to design a kernel that adequately encapsulates information necessary for prediction." The kernel construction process often must leverage the intuition and insights of the designers (e.g., Tan and Wang 2004). As stated earlier, prior research has articulated the need for considering contextual and content-based aspects of documents when constructing appropriate similarity measures for linked document classification (Calado et al. 2006). Accordingly, the page level similarity score Sim(a, k) in the kernel function utilizes structural and content attributes (i.e., body text, source code, URLs, images, etc.) of the pages being compared. Scores are based on the occurrence of the aforementioned set of fraud cues in a and k, as well as the two pages' levels and number of in and out links. The feature vectors $a_1, \ldots a_n$ and $k_1, \ldots k_n$ encompass all elements of the extended fraud cue set, with the exception of in/out links and page levels, since those are measured separately in the context component of Sim(a, k). The tunable parameter λ is used to weight the impact of context and content on Sim(a, k). After testing several different values on the training data using cross-validation, λ was set to 0.5.

The Sim(a, k) is on a 0–1 scale for a given web page k in b, with a score of 1 suggesting that a and k are identical. For each a–b comparison, this results in a vector of similarity scores comprised of m elements (one for each k in b). Next, the average and max similarity score is computed. The average similarity score $\text{Sim}_{\text{ave}}(a, b)$ is the average across all scores in the vector, while the max similarity $\text{Sim}_{\text{max}}(a, b)$ is simply the highest similarity score in the vector (Figure 3).

Table 2. Examples of Fraud Cues Incorporated in AZProtect

Category	Attribute Group	Fraud Cues	Fake Site Type	Description
Web page text	Word phrases	"member FDIC" "about FDIC"	Concocted	References to Federal Deposit Insurance Corporation rarely appear in concocted bank websites.
		"© 2000-2006"	Concocted	Outdated copyrights often appear in concocted websites.
		"fee calculator"	Concocted	Concocted cargo delivery websites provide competitive phony estimates to lure customers. Legitimate sites typically offer estimates in-person through sales representatives.
		"pay by phone" "call toll free"	Concocted	Fraudsters prefer to engage in online transactions. They rarely offer phone-based payment options.
		"payment history" "password management" "enter your account"	Concocted	Concocted websites do not provide considerable support for returning customers since they generally do not have any.
	Lexical measures	Average sentence length	Concocted	Sentences in concocted websites tend to be two to three times longer than ones in legitimate sites.
		Average word length, frequency of long words		Concocted websites often contain concatenated words (e.g., "groundtransport" and "safebankingcenter"), resulting in unusually lengthy words.
		Average number of words per page		Concocted website pages are more verbose than legitimate sites—containing twice as many words per page, on average.
	Spelling and grammar	"Adobe Acrobar"	Concocted	Concocted web pages contain many misspellings and grammatical mistakes.
		"frauduluent"		
		"recieve the"		
		"think forwarder"		
URLs	URL text	"HTTPS"	Concocted, Spoof	Fake websites rarely use the secure sockets layer protocol.
		Random characters in URLs (e.g., "agkd-escrow," "523193pay"	Concocted, Spoof	Since fake websites are mass produced, they use random characters in URLs. It also allows new fake websites to easily circumvent lookup systems that rely on blacklists of exact URLs.
		Number of slashes "/" in URL	Spoof	Spoof sites often piggy back off of legitimate websites or third party hosts. The spoofs are buried deep on these websites' servers.
	Anchor Text	Errors in the URL descriptions (e.g "contactus")	Concocted	Anchor text is used to describe links in web pages. Concocted websites occasionally contain misspelled or inaccurate anchor text descriptions.
Source Code	HTML and Javascript commands	"METHOD POST"	Concocted, Spoof	This HTML command is used to transmit data. It often appears in fake pages that are unsecured (i.e., "HTTP" instead of "HTTPS").
		Image Preloading	Concocted, Spoof	This Javascript code, which is used to preload images to decrease page loading times, rarely appears in fake websites.
	Coding style	"//*" "<!" " =" "//..//"	Concocted, Spoof	Stylistic and syntactic elements in the source code can help identify automatically generated fake websites.
Images	Image meta data	File name, file extension/format, file size	Concocted, Spoof	Fake websites often reuse images from prior fake websites. The file names, extensions, and file sizes can be used to identify duplicate images.
	Image pixels	Pixel colors	Concocted, Spoof	If the image file name and format have been altered, image pixel colors can be used to identify duplicates.
Linkage	Site level	Number of in/out links	Concocted, Spoof	Legitimate websites can contain links to and from many websites, unlike concocted and spoof sites.
	Page level	Number of links, number of relative/absolute links	Concocted, Spoof	Fake websites tend to have fewer pages, and consequently, less linkage between pages. They also often use relative link addresses.

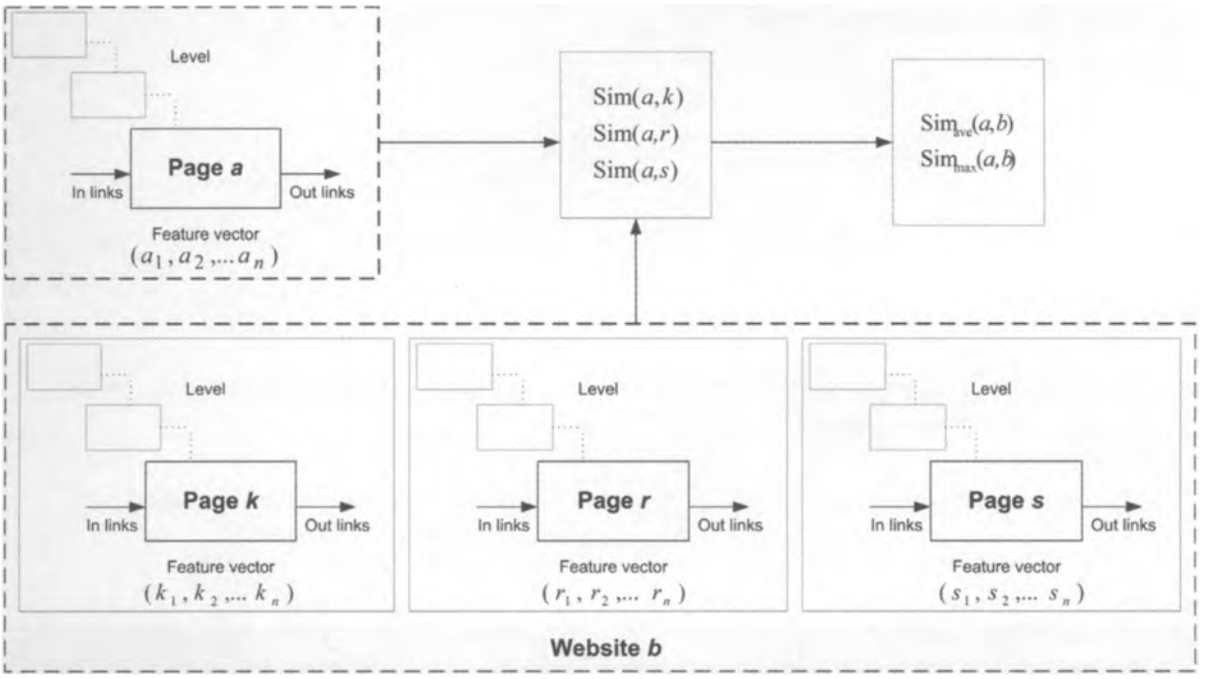


Figure 3. Illustration of Page–Page and Page–Site Similarity Scores Used in the Linear Composite Kernel Function

Repeating this process for each b in the training data results in two page-site similarity vectors for each web page a; x_a consists of average similarities while y_a encompasses max similarities. The inner product of the vectors between every two web pages is computed to produce a kernel matrix of holistic page-page similarity scores that is used as input into the SVM. Figure 4 shows the mathematical formulation for the linear composite kernel.

The kernel is designed to exploit the commonalities between various legitimate websites and between fake websites. Hence, legitimate web pages should be more similar to the pages of other legitimate websites, while fake web pages are likely to have high similarity scores with other fake web pages. Average similarity is intended to capture the overall stylistic similarity between a and web pages appearing in b. Max similarity is designed to measure content duplication. If a web page a copies content from a single page in b, $\text{Sim}_{ave}(a, b)$ may be low if a does not have similarities with the other pages in b. However, $\text{Sim}_{max}(a, b)$ will be high since at least one page in b will contain similar content to a. In contrast, if a features similar visual design elements with pages in b, $\text{Sim}_{ave}(a, b)$ may be high even though a does not copy content from any page in b (i.e., $\text{Sim}_{max}(a, b)$ is low). Utilization of average and maximum similarity enables the consideration of common patterns (via the average similarity score) as well as content duplication (via the max similarity score) that may occur across websites. Collectively, the various characteristics of the custom kernel are intended to enable a more holistic representation of the stylistic tendencies inherent across fake websites.

Figure 5 shows a partial application of the kernel function. In the example, two web pages, a legitimate bank login page and a spoof login page, are compared against four websites (two legitimate and two fake). It is important to note that this is an abbreviated example. In actuality, the kernel would compute the vectors of similarity scores for these web pages in comparison with all training websites (not just these four). Furthermore, the site-level classification would be performed once all web pages belonging to these two websites had been evaluated in a similar fashion.

The top half of the figure shows the in/out link and page-level information for two web pages. For instance, the fake web page is at level 10 (i.e., it is 10 folders deep in the file directory) and has 22 out links (4 of which are listed). A few sample values from the feature vectors of the two web pages are also shown. For example, the URL token "jsp" occurs with a normalized frequency of 0.018966 in the legitimate web page, but doesn't appear in the fake one.

Represent each page a with the vectors:

$x_a = \{\text{Sim}_{\text{ave}}(a, b_1), ..., \text{Sim}_{\text{ave}}(a, b_p)\}$; $y_a = \{\text{Sim}_{\text{max}}(a, b_1), ..., \text{Sim}_{\text{max}}(a, b_p)\}$

Where:

$$\text{Sim}(a,k) = \lambda\left(\left(1 - \frac{|\text{lv}_a - \text{lv}_k|}{\text{lv}_a + \text{lv}_k}\right) + \left(1 - \frac{|\text{in}_a - \text{in}_k|}{\text{in}_a + \text{in}_k}\right) + \left(1 - \frac{|\text{out}_a - \text{out}_k|}{\text{out}_a + \text{out}_k}\right)\right) + (1-\lambda)\left(1 - \frac{1}{n}\sum_{i=1}^{n}\frac{|a_i - k_i|}{a_i + k_i}\right)$$

$$\text{Sim}_{\text{ave}}(a,b) = \frac{1}{m}\sum_{k=1}^{m}\text{Sim}(a,k)$$

$$\text{Sim}_{\text{max}}(a,b) = \underset{k \in \text{pages in site b}}{\arg\max}\ \text{Sim}(a,k)$$

For:

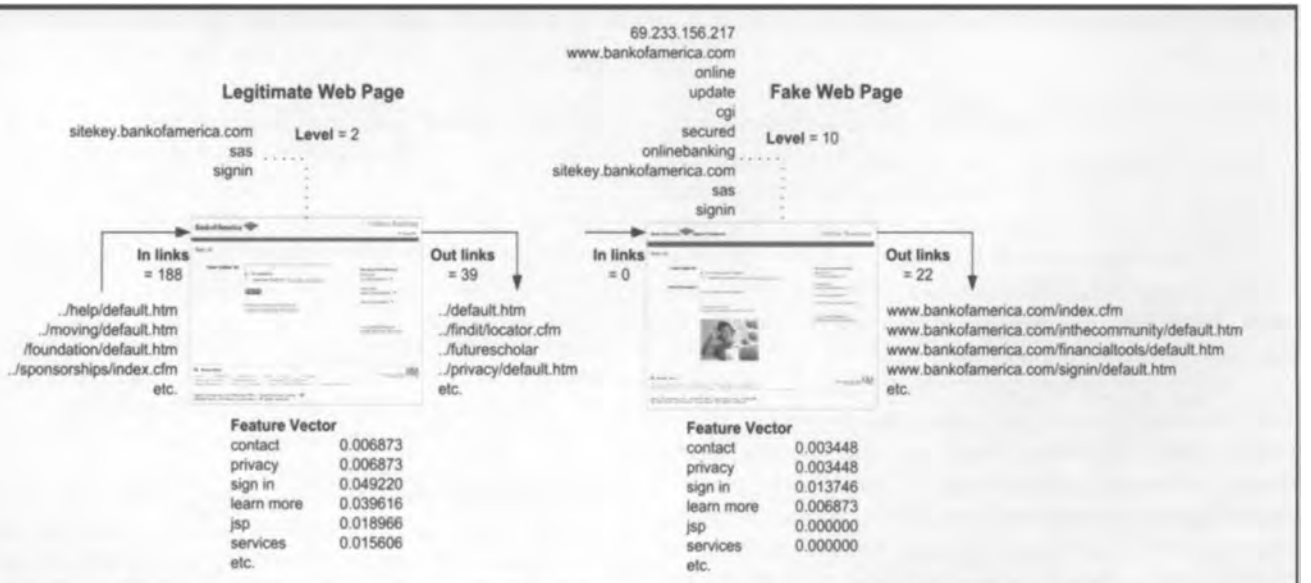
$b \in p$ web sites in the training set; $k \in m$ pages in site b; $a_1 ... a_n$ and $k_1, ... k_n$ are page a and k's feature vectors;

lv_a, in_a, and out_a are the page level and number of in/out links for page a;

The similarity between two pages is defined as the inner product between their two vectors x_1, x_2 and y_1, y_2:

$$\text{Linear Composite Kernel: } K(x_1 + y_1, x_2 + y_2) = \frac{\langle x_1, x_2 \rangle}{\sqrt{\langle x_1, x_1 \rangle \langle x_2, x_2 \rangle}} + \frac{\langle y_1, y_2 \rangle}{\sqrt{\langle y_1, y_1 \rangle \langle y_2, y_2 \rangle}}$$

Figure 4. Linear Composite SVM Kernel for Fake Website Detection



Category	Comparsion Websites	Legitimate Bank Web Page		Spoof Bank Web Page	
		$\text{Sim}_{\text{ave}}(a, b)$	$\text{Sim}_{\text{max}}(a, b)$	$\text{Sim}_{\text{ave}}(a, b)$	$\text{Sim}_{\text{max}}(a, b)$
Legitimate	www.bankofamerica.com	0.495098	1.000000	0.314208	0.392708
	www.citibank.com	0.387834	0.577704	0.247735	0.386828
Fake	beliebo.nl\media\2007\Paypal	0.317838	0.383798	0.522842	0.555874
	65.85.238.84\ebay.monitors\store.ebay	0.257576	0.297611	0.510915	0.577157

Figure 5. Comparing Two Web Pages Against Legitimate and Fake Websites

The bottom half of Figure 5 shows the average and max similarity scores for these two web pages in comparison with the four training websites. As intended, the legitimate web page has higher scores when compared against legitimate websites, while the fake page has higher scores in comparison with fake websites. This is attributable to the stark differences between the legitimate and fake web pages with respect to their number of in links, page levels, and feature vector values. While the number of out links is slightly more comparable, the out links of the fake web pages are all external ones, pointing to web pages in the legitimate website that it is spoofing. The legitimate web page has a max similarity of 1.000000 when compared to the actual Bank of America website, which means that it is identical to one of the web pages from that site (which is to be expected since this page belongs to that website). The example illustrates how the kernel is able to accurately assess legitimate and fake web pages.

The AZProtect System Interface

The AZProtect system interface is comprised of six components (shown in Figure 6). For a user-specified website URL, the system analyzes and categorizes individual web pages in the site as real or fake. For each web page, fraud cues are extracted and the page is classified using the SVM linear composite kernel. The overall score for the website is displayed as the percentage of its web pages deemed fake (component 1 of Figure 6). This score can be construed as an indicator of the system's level of confidence with a particular website's classification. In Figure 6, the website aliantz-del.bravehost.com contains eight web pages, each of which were classified as fake. The web page URLs, and their classifications are shown in the table depicted in component 2 of Figure 6. Pages in the table are color coded as green (for pages considered legitimate) and red (for ones classified as fake). Users can click on any web page in the site; in Figure 6, the selected page (index2.html) is displayed in component 3. The top right panel shows the website's page and site level linkage (e.g., component 4 shows the page level linkage for aliantz-del.bravehost.com). Blue nodes signify pages belonging to this website while gray nodes are pages from other websites. The lines between nodes represent linkage. The node for the selected web page (index2.html) is highlighted in pink. This particular website only links to 24 pages from other websites, a fairly small number for legitimate commercial sites. A table displays the website's key fraud cues related to page text, URLs, and source code elements (component 5). For each fraud cue, the category, cue description, and occurrence frequency are displayed. Image-based fraud cue information is shown in the bottom right panel (component 6). The left-hand portion of the panel displays all of the images appearing in this website, along with the number of prior legitimate and fake websites in which that image has appeared. For example, the selected image, group2.jpg, has appeared in 0 legitimate and 28 fake websites (across our training data set of 1,000 websites). This image, which shows five purported company employees, is displayed on the top right side of the panel. The bottom right side of the panel lists URLs for the 28 fake web pages where the image previously appeared. Although component 1 sufficiently conveys the system's classification results, the additional components are included to help illustrate why a particular website is considered legitimate or fake. The ancillary components display the fraud cues identified in the website, providing users with justification for the classification results.

Evaluation

In the previous section, we discussed the construction of two proposed IT artifacts: the AZProtect system and its SLT-based core classification algorithm, a linear composite SVM kernel function. The evaluation phase is intended to assess existing artifacts and inform the future search process (Hevner et al. 2004; Nunamaker 1992). Accordingly, we conducted a series of experiments to assess the effectiveness of our proposed IT artifacts. Experiment 1 evaluated the set of extended fraud cues utilized in AZProtect in order to confirm the importance of using richer fraud cues. Experiment 2 assessed the effectiveness of AZProtect in comparison with existing fake website detection systems (H1 and H2). Experiment 3 tested the efficacy of using SLT-based learning algorithms over other learning methods (H3), while experiment 4 evaluated the performance of the proposed linear composite kernel against generic kernels that do not incorporate domain-specific knowledge.

We evaluated 900 websites over a 3 week period. The test bed encompassed 200 legitimate, 350 concocted, and 350 spoof websites. The spoof websites were taken from two popular online trading communities: Phishtank.com and the Anti-Phishing Working Group. The spoofs analyzed were replicas of legitimate websites such as eBay, Paypal, Escrow.com, bank and university websites, search engines, etc. The concocted websites were taken from Artists-Against 4-1-9 and Escrow-fraud.com. These included websites pertaining to shipping, financial, escrow, legal, and retail genres. The 200 real websites included ones that are commonly spoofed, as well as those belonging to genres relevant to the concocted website test bed. There was no overlap between the 1,000 websites used to extract fraud cues and train the

Figure 6. Screenshot of the AZProtect System Interface

SVM classifier in AZProtect, and the 900 websites incorporated in our test bed. The experimental design was consistent with prior research (Ntoulas et al. 2006; Zhang et al. 2007).

Unlike the natural sciences, which are concerned with how things are, design science is concerned with how things ought to be, with devising artifacts to attain goals (Simon 1996). Therefore, the *evaluation* of the proposed artifacts enforces value judgments that are typically manifested in the form of utility functions (Simon 1996). We deployed established performance measures to evaluate the quality of the devised fake website detection system and its components. The metrics chosen were closely related to the hypotheses discussed earlier. These included overall accuracy, class-level precision, and class-level recall. Additionally, class-level f-measure and receiver operating characteristic plots/curves were also employed. Class-level f-measure is the harmonic mean of precision and recall. Receiver operating characteristic (ROC) plots/curves depict the relationship between true positive and false positive rates for the various systems and comparison classification techniques.

Comparison of Fraud Cue Categories

In order to ensure that the extended set of fraud cues utilized by AZProtect did in fact result in enhanced fake website detection over individual fraud cue categories, a performance comparison was made. We compared the complete set of 6,000 fraud cues against 5 subsets: web page text, URL, source code, image, and linkage attributes. The comparison was run on the 900-website test bed using AZProtect's SVM classifier with the proposed linear composite kernel. The experimental results are shown in Figure 7.

The extended feature set had the best results in terms of overall accuracy as well as concocted and spoof site detection

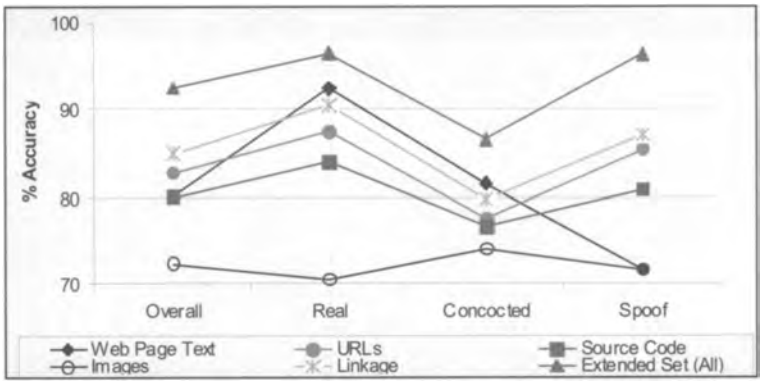


Figure 7. Results for Fraud Cue Comparison

rates. It outperformed all five comparison fraud cue sets by a wide margin, generally between 5 and 10 percent. The results support the notion that an extended set of fraud cues are instrumental in improving fake website detection capabilities. This extended fraud cue set was used in all ensuing experiments involving AZProtect (discussed below).

Comparison of Classifier and Lookup Systems for Fake Website Detection

We evaluated the effectiveness of AZProtect in comparison with other existing fake website detection systems. The comparison tools were seven systems that had either performed well in prior testing or not been evaluated (Zhang et al. 2007). In addition to AZProtect, the comparison systems featured three other classifier systems: SpoofGuard, Netcraft, and eBay's Account Guard. There were also four lookup systems: IE Phishing Filter, FirePhish, EarthLink Toolbar, and Sitehound. The lookup systems all utilized server-side blacklists that were updated regularly by the system providers. In order to allow a fair comparison, all eight systems were evaluated simultaneously on different (but similarly configured) machines. Different machines were necessary in order to avoid undesirable interactions between the comparison systems (i.e., some of the systems did not work well when simultaneously installed on the same machine). All systems classified each of the 900 test bed websites as legitimate or fake. As previously stated, AZProtect uses a predefined page-level classification threshold to categorize websites as legitimate or fake. For the seven comparison systems, thresholds yielding the best results for each respective system were used. For instance, lookup system blacklists seldom contain comprehensive representations of a fake website's URLs (Zhang et al. 2007). Therefore, for the comparison lookup systems, we considered a website fake if any of its pages appeared in the blacklist. This assumption offered the best performance for all four lookup systems. Similarly, the thresholds for the three comparison rule-based systems were retrospectively set in order to allow the highest overall accuracy values.

Table 3 shows the evaluation results. AZProtect had the best overall accuracy and class-level f-measures, in addition to the best recall (i.e., detection rates) on concocted and spoof websites. It outperformed comparison systems by 10 to 15 percent. Netcraft and Spoofguard also performed well, with overall accuracies over 70 percent. The eBay Account Guard classifier system performed poorly in detecting concocted websites (with a recall value of around 3 percent) since it is geared toward identifying eBay and Paypal spoofs.

In comparison with classifier systems, the lookup systems had higher precision on the concocted and spoof websites, with most systems attaining precision values near 100 percent on those two categories. In other words, the lookup systems rarely classified the legitimate websites as fake. This is not surprising, since lookup systems rely on blacklists that are unlikely to contain URLs for legitimate websites (Zhang et al. 2007). However, the lookup systems were particularly weak in terms of their ability to detect fake websites, as evidenced by their low recall values on the concocted and spoof websites. IE Filter and Firephish performed well on spoof sites, but seemed to miss most of the concocted websites. They detected less than 10 percent of the 350 concocted websites in our test bed; a troubling statistic considering the fact that these two filters are incorporated by the two most popular web browsers. Sitehound and the EarthLink toolbar had overall accuracies below 50 percent as well as sub-par recall values on the fake websites, casting serious doubts on the overall usefulness of these systems.

Table 3. Performance Results (%) for Classifier and Lookup Systems

System		Overall Accuracy (n = 900)	Real Websites (n = 200)			Concocted Detection (n = 350)			Spoof Detection (n = 350)		
			F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.
Classifier	AZProtect	**92.56**	**85.21**	**76.29**	96.50	**91.82**	97.74	**86.57**	**97.12**	97.97	**96.29**
	eBay AG	44.89	44.64	28.73	**100.00**	6.09	**100.00**	3.14	71.08	**100.00**	55.14
	Netcraft	83.00	72.13	56.74	99.00	82.28	99.19	70.29	92.52	99.34	86.57
	SpoofGuard	70.00	57.28	41.90	90.50	65.81	90.50	51.71	84.14	93.38	76.57
Lookup	EarthLink	42.67	43.55	27.87	99.50	15.75	96.77	8.57	61.27	99.36	44.29
	IE Filter	55.33	49.87	33.22	**100.00**	17.70	**100.00**	9.71	85.99	**100.00**	75.43
	FirePhish	54.89	49.63	33.00	**100.00**	12.84	**100.00**	6.86	87.09	**100.00**	77.14
	Sitehound	47.33	45.77	29.67	**100.00**	58.59	**100.00**	41.43	37.58	**100.00**	23.14

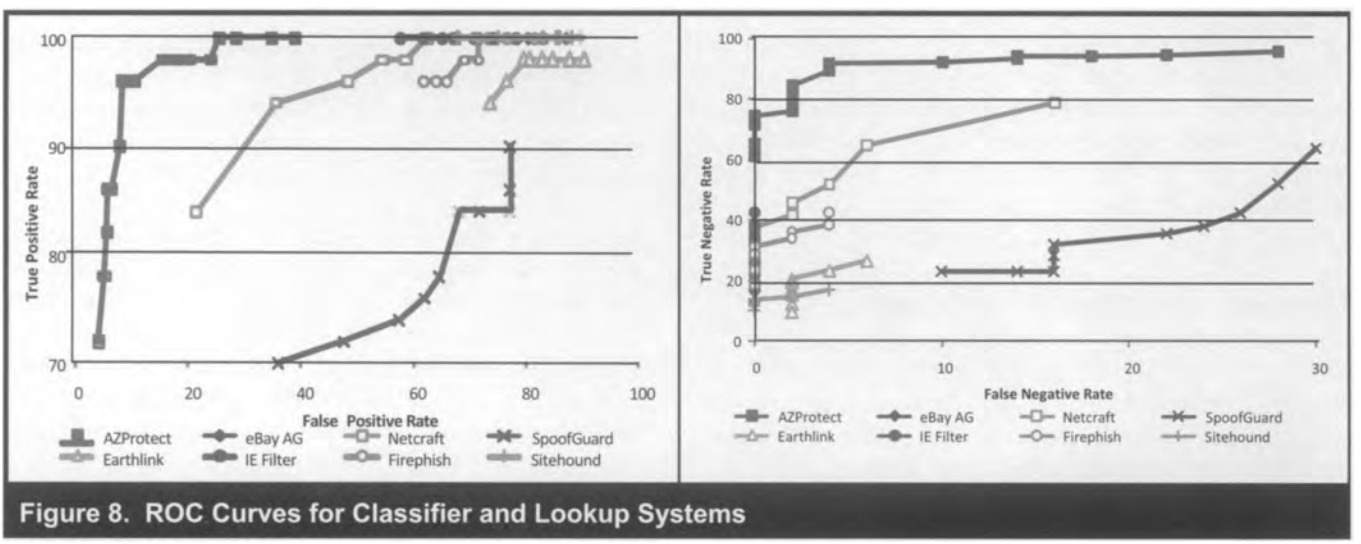


Figure 8. ROC Curves for Classifier and Lookup Systems

Figure 8 depicts the receiver operating characteristic (ROC) curves, showing the tradeoffs between true and false positives as well as true and false negative rates across systems for threshold values ranging from 0.01 to 1.00. Here, a threshold value *t* represents the percentage of web pages that must be classified as fake for the site to be deemed fake. True positives refer to correctly classified legitimate web pages. Curves closer to the top left corner signify better results, since they denote high ratios of true to false positives (or negatives). Looking at the charts, we can see that AZProtect had the best performance, followed by Netcraft. The curves for the lookup systems were clustered in areas with high false positive and low true negative rates, since they classified many fake websites (i.e., ones not in their blacklist) as legitimate. Overall, the results suggest that classifier systems such as AZProtect provide the most desirable combination of true and false positives.

H1: Classifier Versus Lookup Systems

We conducted pair-wise t-tests on overall accuracy (H1a), concocted and spoof recall (H1b), and concocted and spoof precision (H1c). The t-tests compared the performance of our four classifier systems against the four lookup-based tools. This resulted in 16 comparisons for each of our 5 evaluation metrics. Given the large number of comparisons, a Bonferroni correction was performed. Only p-values less than 0.003 were considered to be statistically significant at alpha = 0.05. The t-test results are shown in Table 4.

AZProtect and Netcraft significantly outperformed the four lookup systems in terms of overall accuracy, concocted recall, and spoof recall. SpoofGuard also significantly outperformed all lookup systems in terms of overall accuracy and concocted recall. However, it did not significantly outperform the IE

Table 4. P-Values for Pair Wise t-tests on Classification Accuracy for Classifier Versus Lookup Systems

	H1a – Overall Accuracy							
System	Sitehound	EarthLink	IE Filter	FirePhish				
SpoofGuard	< 0.001	< 0.001	< 0.001	< 0.001				
Netcraft	< 0.001	< 0.001	< 0.001	< 0.001				
EBay AG	0.109*	0.066	< 0.001*	< 0.001*				
AZProtect	< 0.001	< 0.001	< 0.001	< 0.001				
	H1b – Concocted Recall				**H1b – Spoof Recall**			
System	Sitehound	EarthLink	IE Filter	FirePhish	Sitehound	EarthLink	IE Filter	FirePhish
SpoofGuard	0.002	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001	0.340	0.421*
Netcraft	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001
EBay AG	< 0.001*	0.001*	< 0.001*	0.008*	< 0.001	< 0.001	< 0.001*	< 0.001*
AZProtect	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001
	H1c – Concocted Precision				**H1c – Spoof Precision**			
System	Sitehound	EarthLink	IE Filter	FirePhish	Sitehound	EarthLink	IE Filter	FirePhish
SpoofGuard	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001
Netcraft	< 0.001	0.043*	0.006	0.005	0.001	0.359	< 0.001	< 0.001
EBay AG	0.500	0.002*	0.500	0.500	0.500	0.110*	0.500	0.500
AZProtect	< 0.001	0.182*	< 0.001	0.002	0.012	0.090	0.002	0.001

*Result contradicts hypothesis

Filter or FirePhish for spoof recall Due to its limited focus, eBay's Account Guard was significantly outperformed by IE Filter and FirePhish on overall accuracy, concocted recall, and spoof recall. It only managed to significantly outperform Sitehound and EarthLink on spoof recall. Overall, the results seem to support H1a and H1b: classifier systems are better suited than their lookup-only counterparts with respect to overall accuracy and fake website recall. Three of the four classifier systems evaluated significantly outperformed comparison lookup systems on virtually every H1a and H1b condition tested, with the exception being the eBay system. These results are consistent with prior studies that have also emphasized the need for generalizable classification mechanisms in order to achieve suitable recall on fake websites (Zhang et al. 2007).

With respect to concocted and spoof precision, the lookup systems performed better. All four lookup systems significantly outperformed SpoofGuard on concocted and spoof precision. IE, Firephish, and Sitehound also outperformed AZProtect on most conditions. None of the lookup systems outperformed the eBay Account Guard tool, since it tended to be highly precise, albeit with a very limited focus. In general, although the lookup systems yielded high precision values, these results came on very small sets of correctly classified websites (particularly on the concocted websites), rendering many of the precision improvements insignificant. Hence, only 17 of 32 conditions related to H1c were significant.

H2: Learning Versus Rule-Based Classifier Systems

Of the four classifier systems evaluated, only AZProtect uses a learning-based strategy. Spoofguard, Netcraft, and the eBay Account Guard tool all rely on simple rule-based heuristics. We conducted t-tests to assess the effectiveness of our SLT-based classification system in comparison with these three rule-based classifiers (shown in Table 5). AZProtect significantly outperformed all three comparison classification systems in terms of overall accuracy as well as concocted and spoof recall (all nine p-values < 0.001). AZProtect also significantly outperformed SpoofGuard in terms of concocted and spoof precision. However, Netcraft and eBay Account Guard had better fake website precision, although the improvement in eBay Account Guard was not significant. Overall, the results support H2a and H2b. The ability of the SLT-based system to incorporate a rich set of fraud cues facilitated enhanced overall accuracy and better fake website detection rates than existing rule-based classifier systems.

Table 5. P-Values for Pair Wise t-tests on Classification Accuracy for Learning Versus Rule-Based Systems

Hypothesis	Comparison Classifier System		
	SpoofGuard	Netcraft	EBay AG
H2a – Overall Accuracy	< 0.001	< 0.001	< 0.001
H2b – Concocted Recall	< 0.001	< 0.001	< 0.001
H2b – Spoof Recall	< 0.001	< 0.001	< 0.001
H2c – Concocted Precision	< 0.001	< 0.001*	0.081*
H2c – Spoof Precision	< 0.001	0.003*	0.064*

*Result contradicts hypothesis

Comparison of Learning Classifiers for Fake Website Detection

In order to evaluate the classification effectiveness of the proposed SLT-based algorithm, we compared it with several other learning methods, including logistic regression, J48 decision tree, Bayesian network, naïve Bayes, neural network, and Winnow. Each of these comparison techniques has been applied to related classification problems, including text, style, and website categorization (Dumais et al. 1998; Ntoulas et al. 2006; Salvetti and Nicolov 2006; Zheng et al. 2006). However, none of these methods supports the use of custom kernel functions. Each comparison method was trained on the same set of 1,000 websites and extended fraud cues as those used by the SVM classifier in AZProtect. The page-level classification thresholds for the six comparison learning techniques were set retrospectively to enable the best possible site-level classification results.

Table 6 shows the experimental results. SVM outperformed all six comparison methods in terms of overall accuracy as well as class-level f-measures and precision. Furthermore, SVM had the highest spoof detection rate (i.e., recall). It also outperformed the logistic regression, Bayesian network, naïve Bayes, neural network, and Winnow classifiers in terms of recall on the concocted websites. However, J48 was more effective than SVM in terms of its ability to detect concocted websites, with 3 percent higher recall.

In order to assess the impact of the different page-level classification thresholds on site-level performance, we constructed ROC curves (depicted in Figure 9). The curves show the true–false positive and negative rates across learning technique for threshold values ranging from 0.01 to 1.00. Looking at the charts, we can see that SVM had the best performance since its curves were situated closest to the top left corner (the region associated with a high ratio of true to false positives/negatives). Logistic regression, J48 decision trees, and Bayesian network also performed well, while naïve Bayes, Winnow, and neural network lagged behind.

H3: SLT-Based Learning Classifier Versus Other Learning Classifiers

Table 7 shows the t-test results comparing SVM against comparison techniques. P-values less than 0.05 were considered significant. SVM significantly outperformed the six other learning classifiers in terms of overall accuracy (all six p-values < 0.001). It also outperformed the comparison techniques on concocted and spoof recall for 9 out of the 12 t-test conditions. However, it did not significantly outperform logistic regression and Bayesian network on concocted recall. J48 also had better results for concocted recall. Nevertheless, the competitive performance of the J48 decision tree, Bayesian network, and logistic regression method were negated on the legitimate websites, where SVM outperformed them by 6 to 18 percent. The high false positive rates of the comparison classifiers resulted in SVM significantly outperforming them in terms of concocted and spoof precision. The t-test results support H3a, H3b, and H3c and suggest that SLT-based classifiers with enhanced generalization power and domain-specific kernel functions can provide improved fake website detection results over non-kernel based learning methods. These results are consistent with classification results in related domains, where SVM has also been shown to outperform comparison methods (Dumais et al. 1998; Zheng et al. 2006).

Table 6. Performance Results (%) for Various Learning-Based Classification Techniques

Learning Technique	Overall Accuracy (n = 900)	Real Websites (n = 200)			Concocted Detection (n = 350)			Spoof Detection (n = 350)		
		F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.
SVM	**92.56**	**85.21**	**76.29**	**96.50**	**91.82**	**97.74**	86.57	**97.12**	**97.97**	**96.29**
Logistic regression	89.00	78.53	69.36	90.50	90.02	94.08	86.29	92.58	94.36	90.86
J48 Decision Tree	88.77	75.66	73.01	78.50	88.82	87.95	**89.71**	90.98	88.41	93.71
Bayesian Network	88.56	77.27	69.18	87.50	88.72	92.28	85.43	92.55	92.82	92.29
Naïve Bayes	77.67	63.12	49.86	86.00	86.49	91.14	82.29	77.47	89.51	68.29
Winnow	76.11	58.73	47.66	76.50	80.96	85.17	77.14	79.52	84.79	74.86
Neural Network	66.22	54.21	38.79	90.00	70.63	90.99	57.71	73.28	91.45	61.13

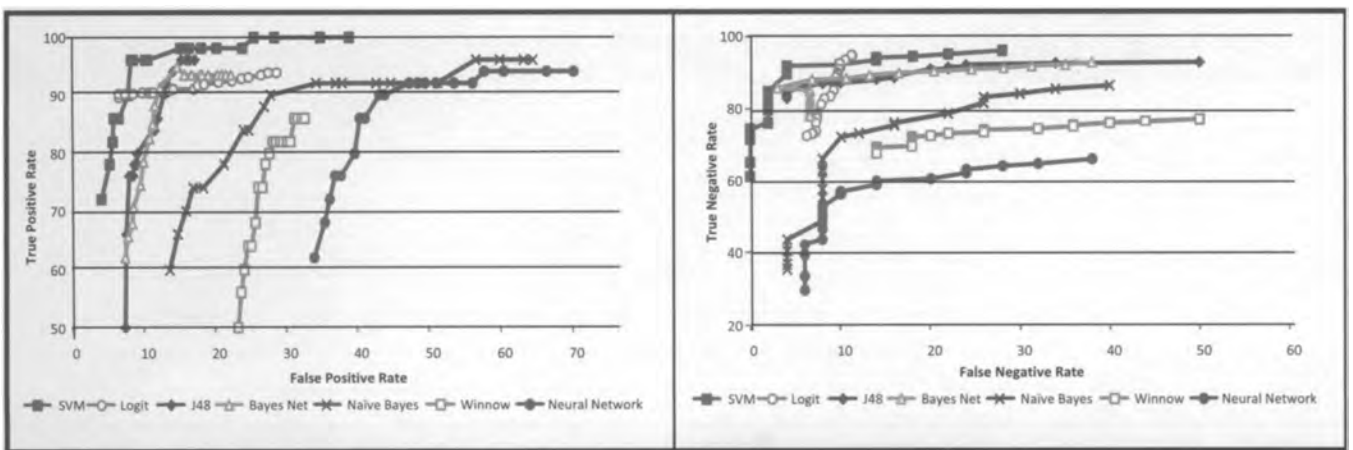


Figure 9. ROC Curves for Various Learning Classifiers

Table 7. P-Values for Pair Wise t-tests on Classification Accuracy for SVM Versus Alternative Learning Techniques

Hypothesis	Comparison Learning Techniques					
	Logit	J48	Bayes Net	Naïve Bayes	Neural Net	Winnow
H3a – Overall Accuracy	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001
H3b – Concocted Recall	0.426	0.051*	0.083	< 0.001	< 0.001	< 0.001
H3b – Spoof Recall	< 0.001	0.030	< 0.001	< 0.001	< 0.001	< 0.001
H3c – Concocted Precision	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001
H3c – Spoof Precision	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001

*Result contradicts hypothesis

Table 8. Performance Results (%) for Various Kernel Functions

Learning Technique	Overall Accuracy (n = 900)	Real Websites (n = 200)			Concocted Detection (n = 350)			Spoof Detection (n = 350)		
		F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.
Linear Custom Composite	**92.56**	**85.21**	**76.29**	**96.50**	**91.82**	**97.74**	**86.57**	**97.12**	**97.97**	**96.29**
Linear Equal Weights	90.78	82.00	72.42	94.50	90.61	96.45	85.43	95.36	96.76	94.00
Polynomial 2nd Degree	90.44	81.38	71.75	94.00	90.30	96.13	85.14	95.07	96.47	93.71
Radial Basis Function	90.22	81.20	70.90	95.00	90.41	96.74	84.86	94.89	97.02	92.86
Polynomial 3rd Degree	89.89	80.60	70.26	94.50	88.58	96.31	82.00	95.96	96.80	95.14
Linear Information Gain	87.33	76.44	65.14	92.50	87.86	95.02	81.71	92.65	95.45	90.00

Table 9. P-Values for Pair Wise t-tests on Classification Accuracy for Custom and Comparison Kernel Functions

Hypothesis	Comparison Kernel Functions				
	Linear Equal	Polynomial 2nd	Radial Basis	Polynomial 3rd	Linear Information
H4a – Overall Accuracy	0.042	0.028	0.040	0.014	< 0.001
H4b – Concocted Recall	0.311	0.225	0.016	< 0.001	< 0.001
H4b – Spoof Recall	0.029	0.017	0.026	0.1392	< 0.001
H4c – Concocted Precision	0.164	0.004	0.022	< 0.001	< 0.001
H4c – Spoof Precision	0.028	0.010	0.035	0.011	< 0.001

Comparison of Kernel Functions for Fake Website Detection

An important element of the AZProtect system is its custom linear composite kernel. This kernel was customized for representing important characteristics of fake websites. We evaluated its effectiveness in comparison with kernel functions that do not incorporate problem-specific characteristics related to the fake website domain. These kernels were applied to the same set of fraud cues as the custom linear composite kernel. The comparison kernels employed included a linear kernel that weighted all attributes in the input feature vectors equally (Ntoulas et al. 2006), as well as a linear kernel that weighted each attribute in the feature vector based on its information gain score (attained on the training data). Additionally, second and third degree polynomial kernels and a radial basis function kernel were incorporated (Drost and Scheffer 2005). As with previous experiments, the page-level threshold yielding the best site-level overall accuracy results was used for each of the comparison kernels. Table 8 shows the experimental results. The linear composite kernel outperformed all five comparison kernels in terms of overall accuracy as well as class-level f-measures, precision, and recall on real, concocted, and spoof websites. The linear composite kernel was approximately 2 percent to 5 percent better than the comparison kernels on most evaluation metrics. Comparing our results with prior work that investigated spoof website detection using generic linear kernels (e.g., Pan and Ding 2006), our overall accuracy as well as legitimate and spoof website recall were 8 percent to 10 percent higher. Additionally, the proposed linear composite kernel had over 10 percent lower false negative rates (i.e., legitimate websites deemed fake). This stark improvement was attributable to the use of a richer set of fraud cues and the custom linear composite kernel.

H4: Custom Linear Composite Kernel Versus Other Kernels

Table 9 shows the t-test results comparing the linear composite kernel against comparison kernels. The linear com-

posite kernel significantly outperformed the five other kernels in terms of overall accuracy, with all p-values less than 0.05. It also significantly outperformed the comparison kernels on 7 out of 10 conditions pertaining to concocted and spoof recall. However, the concocted recall performance gain over the linear equal weight and polynomial second degree kernels was not significant. With respect to H4c, the linear composite kernel significantly outperformed the comparison kernels on all but one condition (the linear equal weight kernel on concocted precision). Interestingly, the results provide strong support for H4a, H4b, and H4c, even though the performance margins were not as wide as those in previous experiments (e.g., those in the two previous sections). Deeper analysis of the erroneous classifications for the different kernels revealed that the linear composite kernel was able to dominate the other kernels in many cases. That is, in addition to correctly classifying most of the websites also correctly classified by the other kernels, the linear composite kernel was able to accurately categorize 20 to 40 more websites (which were missed by the other kernels). The results support the concept that kernel functions incorporating problem-specific domain knowledge are capable of attaining higher accuracy as well as better fake website precision and recall rates.

Results Discussion

Earlier, we identified four important design characteristics necessary for any effective fake website detection system. First, detection systems must be able to generalize across fake website categories. The results in experiment 2 (and H1) demonstrated the enhanced generalization ability of classifier systems in comparison with those relying on lookup mechanisms. Classifier systems were able to perform well on concocted and spoof sites, while most lookup systems faltered in their ability to detect concocted websites.

Given the complex nature of fake websites, a second compulsory characteristic for detection systems is that they must be scalable in terms of the number of fraud cues utilized. The proposed AZProtect classifier system outperformed its rule-based counterparts (as demonstrated by H2). This was attributable to the use by those systems of a small set of manually observed rules that are simply not sufficient to capture the spectrum of tendencies exhibited by fake websites (Zhang et al. 2007). The ineffectiveness of rule-based systems was especially evident on the concocted websites, which seem to be more difficult to detect. Performance by Spoofguard and Netcraft fell between 15 and 25 percent on the concocted websites, as compared to their spoof detection rates. In contrast, AZProtect's performance fell by a smaller margin since the system uses a rich set of fraud cues.

In experiment 3, comparison learning methods were surpassed by the SLT-based SVM classifier in terms of overall accuracy and fake website detection rates (H3). The augmented generalization power of SVM and its ability to incorporate specific properties of fake websites contributed to its improved performance. The linear composite kernel function utilized considers stylistic commonalities and content duplication across websites, while using a representation that preserves the relationships across input objects. Experiment 4 further demonstrated usefulness of the proposed kernel over those that do not incorporate specific characteristics useful for detecting fake websites (H4). Follow-up experiments revealed that retraining the proposed linear composite SVM kernel as newer fake website samples became available further improved overall accuracy as well as class-level precision and recall. Although these experiments could not be included due to space limitations, they displayed the potential adaptability of dynamic SLT-based classifiers, an essential characteristic given the evolving nature of fake websites. Collectively, the results lend credence to the notion that SLT-based fake website detection systems could dramatically reduce the effectiveness of fake websites.

While the results are very promising, usability studies are needed to assess the effectiveness of such systems in practice. These studies must address a bevy of issues, including appropriate interface design choices, methods for enhancing the perceived usefulness of the system, alternatives for improving users' confidence in the results of the systems, etc. Currently, the most commonly used fake website detection instruments are the built-in filters that accompany the IE and Firefox web browsers (Li and Helenius 2007; Zhang et al. 2007). The lack of appropriate justification for classifications and poor results of these systems have hindered their adoption, usefulness, and authority in the eyes of end users. Users are not very trusting of their recommendations (Wu et al. 2006). The SLT-based classifier system developed in this study outperformed those two systems by nearly 40 percent in terms of overall accuracy; the margin was even larger on the fake website subset (Table 3). Hence, the widespread usage of more effective fake website detection systems (i.e., more accurate systems that are also capable of effectively disclosing the logic behind their recommendations), such as the SLT-based one proposed in this work, could dramatically reduce the high rates of return-on-fraud currently enjoyed by fraudsters.

Conclusions

Important design science contributions may include design principles, and a demonstration of the feasibility of those

principles (Hevner et al. 2004). In this research, we proposed a set of design guidelines which advocated the development of SLT-based classification systems for fake website detection. The AZProtect system served as a proof-of-concept, portraying the efficacy of our design guidelines. In addition to these guidelines, the two resulting IT artifacts themselves also constitute research contributions: the AZProtect system and its underlying linear composite SVM kernel. The guidelines and system presented are obviously not without their shortcomings. For instance, in this study we used a single classifier scheme that differentiated legitimate websites from fake ones. However, there are certainly some differences between spoof and concocted websites attributable to their underlying fraud mechanisms utilized. Multiple-classifier schemes (e.g., hierarchical classification models or meta-learning strategies) that employ separate classifiers for spoof and concocted websites could conceivably result in further improvements to the performance of SLT-based classifiers. Nevertheless, this study is the first to propose a systematic framework for the development of fake website detection systems. Therefore, the process of design, development, and evaluation undertaken in this study are of particular importance to developers of online security systems. We are hopeful that future research and practice regarding fake website detection systems will build upon this endeavor.

This study has important implications for several audiences. We presented an in depth look into the fake website problem, including properties of fake websites, methods available to counter them, and the challenges associated with enhanced detection. This discussion is highly relevant to Internet users, IT security practitioners, and firms engaging in e-commerce. Following the lead of Chua et al. (2007), we too believe that alleviating Internet fraud requires collective action. Online trading communities, Internet users, industry, and government must all work together to develop improved mechanisms for preventing online fraud (Chua and Wareham 2004). The guidelines presented in this study are consistent with this notion. The proposed SLT-based system used fraud cues derived from fake websites taken from online trading community databases powered by individual user's reports. Future systems may also wish to consider incorporating this invaluable user feedback into the classification process. With Internet fraud via fake websites accounting for billions of dollars in fraudulent revenue (Dinev 2006; McGuire 2003), using collective action to develop enhanced fake website detection tools could reap substantial rewards. Considering that the brunt of the impact directly relates to B2C commerce, significantly affecting businesses and consumers, the opportunity cost of inaction is far too severe.

References

Abbasi, A., and Chen, H. 2007. "Detecting Fake Escrow Websites Using Rich Fraud Cues and Kernel Based Methods," in *Proceedings of the 17th Workshop on Information Technologies and Systems*, Montreal, Canada, pp. 55-60.

Abbasi, A., and Chen, H. 2008a. "CyberGate: A Design Framework and System for Text Analysis of Computer-Mediated Communication," *MIS Quarterly* (32:4), pp. 811-837.

Abbasi, A., and Chen, H. 2008b. "Writeprints: A Stylometric Approach to Identity-level Identification and Similarity Detection in Cyberspace," *ACM Transactions on Information Systems* (26:2), Article 7.

Abbasi, A., Chen, H., and Nunamaker Jr., J. F. 2008. "Stylometric Identification in Electronic Markets: Scalability and Robustness," *Journal of Management Information Systems* (25:1), pp. 49-79.

Airoldi, E., and Malin, B. 2004. "Data Mining Challenges for Electronic Safety: The Case of Fraudulent Intent Detection in E-Mails," in *Proceedings of the Workshop on Privacy and Security Aspects of Data Mining*, Los Alamitos, CA: IEEE Computer Society, pp. 57-66.

Arazy, O., and Woo, C. 2007. "Enhancing Information Retrieval Through Statistical Natural Language Processing: A Study of Collocation Indexing," *MIS Quarterly* (31:3), pp. 525-546.

Bayes, T. 1958. "Studies in the History of Probability and Statistics: XI. Thomas Bayes' Essay Towards Solving a Problem in the Doctrine of Chances," *Biometrika* (45), pp. 293-295.

Burges, C. J. C. 1998. "A Tutorial on Support Vector Machines for Pattern Recognition," *Data Mining and Knowledge Discovery* (2), pp. 121-167.

Calado, P., Cristo, M., Goncalves, M. A., de Moura, E. S., Ribeiro-Neto, B., and Ziviani, N. 2006. "Link-Based Similarity Measures for the Classification of Web Documents," *Journal of the American Society for Information Science and Technology* (57:2), pp. 208-221.

Cao, L., and Gu, Q. 2002. "Dynamic Support Vector Machines for Non-Stationary Time Series Forecasting," *Intelligent Data Analysis* (6:1), pp. 67-83.

Chen, W. S., Yuen, P. C., Huang, J., and Dai, D. Q. 2005. "Kernel Machine-Based One-Parameter Regularized Fisher Discriminant Method for Face Recognition," *IEEE Transactions on Systems Man and Cybernetics Part B* (35:4), pp. 659-669.

Chou, N., Ledesma, R., Teraguchi, Y., Boneh, D., and Mitchell, J. C. 2005. "Client-Side Defense Against Web-Based Identity Theft," in *Proceedings of the 11th Annual Network and Distributed System Security Symposium*, San Diego, CA (available online at http://www.isoc.org/isoc/conferences/ndss/04/proceedings/Papers/Chou.pdf).

Chua, C. E. H., and Wareham, J. 2004. "Fighting Internet Auction Fraud: An Assessment and Proposal," *IEEE Computer* (37:10), pp. 31-37.

Chua, C. E. H., Wareham, J., and Robey, D. 2007. "The Role of Online Trading Communities in Managing Internet Auction Fraud," *MIS Quarterly* (31:4), pp. 759-781.

Cristianini, N., and Shawe-Taylor, J. 2000. *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*, New York: Cambridge University Press.

Cyr, D. 2008. "Modeling Web Site Design Across Cultures: Relationships to Trust, Satisfaction, and E-Loyalty," *Journal of Management Information Systems* (24:4), pp. 47-72.

Dinev, T. 2006. "Why Spoofing is Serious Internet Fraud," *Communications of the ACM* (49:10), pp. 76-82.

Drost, I., and Scheffer, T. 2005. "Thwarting the Nigritude Ultramarine: Learning to Identify Link Spam," in *Proceedings of the 16th European Conference on Machine Learning (ECML)*, pp. 96-107.

Duda, R. O., Hart, P. E., and Stork, D. G. 2000. *Pattern Classification* (2nd ed.), New York: Wiley.

Dumais, S., Platt, J., Heckerman, D., and Sahami, M. 1998. "Inductive Learning Algorithms and Representations for Text Categorization," in *Proceedings of the Seventh of ACM Conference on Information and Knowledge Management,* Bethesda, MD, November 2-7, pp. 148-155.

Everard, A. P., and Galletta, D. F. 2005. "How Presentation Flaws Affect Perceived Site Quality, Trust, and Intention to Purchase from an Online Store," *Journal of Management Information Systems* (22:3), pp. 56-95.

Fetterly, D., Manasse, M., and Najork, M. 2004. "Spam, Damn Spam, and Statistics," in *Proceedings of the Seventh International Workshop on the Web and Databases*, Pais, June 17-18, pp. 1-6.

Forman, G. 20003. "An Extensive Empirical Study of Feature Selection Metrics for Text Classification," *Journal of Machine Learning Research* (3), pp. 1289-1305.

Fu, A. Y., Liu, W., and Deng, X. 2006. "Detecting Phishing Web Pages with Visual Similarity Assessment Based on Earth Mover's Distance (EMD)," *IEEE Transactions on Dependable and Secure Computing* (3:4), pp. 301-311.

Gefen, D., and Straub, D. 2003. "Managing User Trust in B2C e-Services," *eService Journal* (2:2), pp. 7-24.

Grazioli, S., and Jarvenpaa, S. L. 2000. "Perils of Internet Fraud: An Empirical Investigation of Deception and Trust with Experienced Internet Consumers," *IEEE Transactions on Systems, Man, and Cybernetics Part A* (20:4), pp. 395-410.

Guyon, I., and Elisseeff, A. 2003. "An Introduction to Variable and Feature Selection," *Journal of Machine Learning Research* (3), pp. 1157-1182.

Gyongyi, Z., and Garcia-Molina, H. 2005. "Spam: It's Not Just for Inboxes Anymore," *IEEE Computer* (38:10), pp. 28-34.

Hariharan, P., Asgharpour, F., and Camp, L. J. 2007. "NetTrust – Recommendation System for Embedding Trust in a Virtual Realm," in *Proceedings of the ACM Conference on Recommender Systems*, Minneapolis, MN.

Hevner, A. R., March, S. T., Park, J., and Ram, S. 2004. "Design Science in Information Systems Research," *MIS Quarterly* (28:1), pp. 75-105.

Hoar, S. B. 2005. "Trends in Cybercrime: The Darkside of the Internet," *Criminal Justice* (20:3), pp. 4-13.

Jagatic, T. N., Johnson, N. A., Jakobsson, M., and Menczer, F. 2007. "Social Phishing," *Communications of the ACM* (50:10), pp. 94-100.

Joachims, T. 2002. *Learning to Classify Text Using Support Vector Machines: Methods, Theory, and Algorithms*, Boston: Kluwer Academic Publishers.

Jøsang, A., Ismail, R., and Boyd, C. 2007. "A Survey of Trust and Reputation Systems for Online Service Provision," *Decision Support Systems* (43:2), pp. 618-644.

Kolari, P., Finin, T., and Joshi, A. 2006. "SVMs for the Blogosphere: Blog Identification and Splog Detection," in *AAAI Spring Symposium on Computational Approaches to Analyzing Weblogs*, Baltimore County, MD.

Koufaris, M. 2002. "Applying the Technology Acceptance Model and Flow Theory to Online Consumer Behavior," *Information Systems Research* (13:2), pp. 205-223.

Krsul, I., and Spafford, H. E. 1997. "Authorship Analysis: Identifying the Author of a Program," *Computer Security* (16:3), pp. 233-257.

Levy, E. 2004. "Criminals Become Tech Savvy," *IEEE Security and Privacy* (2:2), pp. 65-68.

Li, L., and Helenius, M. 2007. "Usability Evaluation of Anti-Phishing Toolbars," *Journal in Computer Virology* (3:2), pp. 163-184.

Littlestone, N. 1988. "Learning Quickly When Irrelevant Attributes are Abound: A New Linear Threshold Algorithm," *Machine Learning* (2), pp. 285-318.

Liu, W., Deng, X., Huang, G., and Fu, A. Y. 2006. "An Anti-phishing Strategy Based on Visual Similarity Assessment," *IEEE Internet Computing* (10:2), pp. 58-65.

Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N., and Watkins, C. 2002. "Text Classification Using String Kernels," *Journal of Machine Learning Research* (2:3), pp. 419-444.

Lowry, P. B., Vance, A., Moody, G., Beckman, B., and Read, A. 2008. "Explaining and Predicting the Impact of Brand Alliances and Web Site Quality on Initial Consumer Trust of E-Commerce Web Sites," *Journal of Management Information Systems* (24:4), pp. 199-224.

MacInnes, I., Damani, M., and Laska, J. 2005. "Electronic Commerce Fraud: Towards an Understanding of the Phenomenon," in *Proceedings of the 38th Hawaii International Conference on Systems Sciences*, Los Alamitos, CA: IEEE Computer Society Press, p. 181a.

Malhotra, N. K., Kim, S. S., and Agarwal, J. 2004. "Internet Users' Information Privacy Concern (IUIPC): The Construct, the Scale, and a Causal Model," *Information Systems Research* (15:4), pp. 336-355.

March, S. T., and Smith, G. 1995. "Design and Natural Science Research on Information Technology," *Decision Support Systems* (15:4), pp. 251-266.

Markus, M. L., Majchrzak, A., and Gasser, L. 2002. "A Design Theory for Systems that Support Emergent Knowledge Processes," *MIS Quarterly* (26:3), pp. 179-212.

McGuire, D. 2003. "FTC, Businesses Renewing Fight Against ID Theft," *Washington Post*, September 3.

McKnight, D. H., Choudhury, V., and Kacmar, C. 2002. "Developing and Validating Trust Measures for E-Commerce," *Information Systems Research* (13:3), pp. 334-359.

Meyer, D., Leisch, F., and Hornik, K. 2003. "The Support Vector Machine Under Test," *Neurocomputing* (55:1-2), pp. 169-186.

Muller, K., Mika, Sebastian, M., Ratsch, G., Tsuda, K., and Scholkopf, B. 2001. "An Introduction to Kernel-Based Learning Algorithms," *IEEE Transactions on Neural Networks* (12:2), pp. 181-201.

Ntoulas, A., Najork, M., Manasse, M., and Fetterly, D. 2006. "Detecting Spam Web Pages through Content Analysis," in *Proceedings of the 15th International World Wide Web Conference*, Edinburgh, Scotland, May 23-26, pp. 83-92.

Nunamaker, Jr., J. F. 1992. "Build and Learn, Evaluate and Learn," *Informatica* (1:1).

Nunamaker, Jr., J. F., Chen, M., and Purdin, T. D. M. 1991. "Systems Development in Information Systems Research," *Journal of Management Information Systems* (7:3), pp.89-106.

Pan, Y., and Ding, X. 2006. "Anomaly Based Web Phishing Page Detection," in *Proceedings of the 22nd Annual Computer Security Applications Conference*, Washington, DC, December 11-15, pp. 381-392.

Pavlou, P. A., and Gefen, D. 2005. "Psychological Contract Violation in Online Marketplaces: Antecedents, Consequences, and Moderating Role," *Information Systems Research* (16:4), pp. 372-399.

Quinlan, R. 1986. "Induction of Decision Trees," *Machine Learning* (1:1), pp. 81-106.

Russell, S. J., and Norvig, P. *2003. Artificial Intelligence: A Modern Approach* (2nd ed.), Upper Saddle River, NJ: Prentice Hall.

Salvetti, F., and Nicolov, N. 2006. "Weblog Classification for Fast Splog Filtering: A URL Language Model Segmentation Approach," in *Proceedings of the Human Language Technology Conference*, New York, June 4-9, pp. 137-140.

Sebastiani, F. 2002. "Machine Learning in Automated Text Categorization," *ACM Computing Surveys* (34:1), pp. 1-47.

Selis, P., Ramasastry, A., and Wright, C. S. 2001. "Bidder Beware: Toward a Fraud-Free Marketplace—Best Practices for the Online Auction Industry," Center for Law, Commerce & Technology, School of Law, University of Washington, April 17.

Shannon, C. E. 1948. "A Mathematical Theory of Communication," *Bell Systems Technical Journal* (27:10), pp. 379-423.

Shawe-Taylor, J., and Cristianini, N. 2004. *Kernel Methods for Pattern Analysis*, New York: Cambridge University Press.

Shen, G., Gao, B., Liu, T. Y., Feng, G., Song, S., and Li, H. 2006. "Detecting Link Spam Using Temporal Information," in *Proceedings of the Sixth International Conference on Data Mining*, Hong Kong, December 18-22, pp. 1049-1053.

Simon, H. A. 1996. *The Sciences of the Artificial* (3rd ed.), Boston, MA: MIT Press.

Sullivan, B. 2002. "Fake Escrow Site Scam Widens: Auction Winners Sometimes Lose $40,000 at a Time," *MSNBC.com*, December 17 (http://www.msnbc.msn.com/ id/3078510/).

Sun, A., Lim, E.-P., Ng, W.-K., and Srivastava, J. 2004. "Blocking Reduction Strategies in Hierarchical Text Classification," *IEEE Transactions on Knowledge and Data Engineering* (16:10), pp. 1305-1308.

Tan, Y., and Wang, J. 2004. "A Support Vector Machine with a Hybrid Kernel and Minimal Vapnik-Chervonenkis Dimension," *IEEE Transactions on Knowledge and Data Engineering* (16:4), pp. 385-395.

Urvoy, T., Lavergne, T., and Filoche, P. 2006. "Tracking Web Spam with Hidden Style Similarity," in *Proceedings of the 2nd International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, Seattle, WA, August 10, pp. 25-31.

Vapnik, V. 1999a. *The Nature of Statistical Learning Theory*, New York: Springer-Verlag.

Vapnik, V. 1999b. "An Overview of Statistical Learning Theory," *IEEE Transactions on Neural Networks* (10:5), pp. 988-999.

Walls, J. G., Widmeyer, G. R., and El Sawy, O. A. 1992. "Building an Information System Design Theory for Vigilant EIS," *Information Systems Research* (3:1), pp. 36-59.

Wu, B., and Davison, B. D. 2006. "Detecting Semantic Cloaking on the Web," in *Proceedings of the 15th International Conference on World Wide Web*, Edinburgh, Scotland, May 23-26, pp. 819-828.

Wu, M., Miller, R. C., and Garfunkel, S. L. 2006. "Do Security Toolbars Actually Prevent Phishing Attacks?," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems,* Montréal, April 22-27, pp. 601-610.

Yu, H., Han, J., and Chang, K. C.-C. 2004. "PEBL: Web Page Classification without Negative Examples," *IEEE Transactions on Knowledge and Data Engineering* (16:1), pp. 70-81.

Zelenko, D., Aone, C., and Richardella, A. 2003. "Kernel Methods for Relation Extraction," *Journal of Machine Learning Research* (3:6), pp. 1083-1106.

Zhang, Y., Egelman, S., Cranor, L., and Hong, J. 2007. "Phinding Phish: Evaluating Anti-Phishing Tools," in *Proceedings of the 14th Annual Network and Distributed System Security Symposium*, San Diego, CA, February 28-March 2.

Zheng, R., Li, J., Huang, Z., and Chen, H. 2006. "A Framework for Authorship Analysis of Online Messages: Writing-Style Features and Techniques," *Journal of the American Society for Information Science and Technology* (57:3), pp. 378-393.

Zhou, S., and Wang, K. 2005. "Localization Site Prediction for Membrane Proteins by Integrating Rule and SVM Classification," *IEEE Transactions on Knowledge and Data Engineering* (17:12), pp. 1694-1705.

About the Authors

Ahmed Abbasi is an assistant professor in the Sheldon B. Lubar School of Business at the University of Wisconsin–Milwaukee. He received his Ph.D. in MIS from the University of Arizona and an M.B.A. and B.S. in Information Technology from Virginia Tech. His research interests include development and evaluation of technologies for enhanced analysis of computer-mediated communication and improved online security. His research has appeared in *MIS Quarterly*, *Journal of Management Information Systems*, *ACM Transactions on Information Systems*, *IEEE Transactions on Knowledge and Data Engineering*, *IEEE Intelligent Systems*, *IEEE Computer*, and *Journal of the American Society for Information Science and Technology*, amongst others. He is a member of the AIS and IEEE.

Zhu Zhang is an assistant professor in the Department of MIS at the University of Arizona. He received his Ph.D. in Computer and Information Science from the University of Michigan. His research interests include data and text mining, machine learning, and Internet computing. His work has appeared in the *Journal of Management Information Systems, Journal of the American Society for Information Science and Technology, IEEE Intelligent Systems, Information Systems*, and various conference proceedings. Zhu is a member of the AIS, AAAI, and ACL.

David Zimbra is a doctoral student in the Department of MIS in the Eller College of Management at the University of Arizona. He received an M.S. and B.S. in Information Systems from Santa Clara University. His research interests include fraud detection, web mining for business intelligence, and time series forecasting.

Hsinchun Chen is McClelland Professor of MIS at the University of Arizona. He received a B.S. from the National Chiao-Tung University in Taiwan, an MBA from SUNY Buffalo, and a Ph.D. in Information Systems from New York University. Dr. Chen is a Fellow of IEEE and AAAS. He received the IEEE Computer Society 2006 Technical Achievement Award and the INFORMS Design Science Award in 2008. He is author/editor of 20 books, 25 book chapters, and more than 200 journal articles covering topics such as digital libraries, data/text/web mining, intelligence analysis, cyber crime, and security informatics. He has been an advisor for major National Science Foundation, Department of Justice, Department of Defense, and Department of Homeland Security programs pertaining to digital library, digital government, and national security research.

Jay F. Nunamaker, Jr., is Regents and Soldwedel Professor of MIS, Computer Science and Communication at the University of Arizona. He received his Ph.D. in Systems Engineering and Operations Research from Case Institute of Technology, an M.S. and B.S. in Engineering from the University of Pittsburgh, and a B.S. from Carnegie Mellon University. Dr. Nunamaker received the LEO Award from the Association of Information Systems in 2002. This award is given for a lifetime of exceptional achievement in information systems. He was elected as a fellow of the Association of Information Systems in 2000. Dr. Nunamaker has over 40 years of experience in examining, analyzing, designing, testing, evaluating, and developing information systems. He has served as a test engineer at the Shippingport Atomic Power facility, as a member of the ISDOS team at the University of Michigan, and as a member of the faculty at Purdue University, prior to joining the faculty at the University of Arizona in 1974.