Enhancing Predictive Analytics for Anti-Phishing by Exploiting Website Genre Information

AHMED ABBASI, FATEMEH “MARIAM” ZAHEDI, DANIEL ZENG, YAN CHEN, HSINCHUN CHEN, AND JAY F. NUNAMAKER JR.

AHMED ABBASI is an associate professor of information technology (IT) and director of the Center for Business Analytics in the McIntire School of Commerce at the University of Virginia. He attained his B.S. and MBA degrees from Virginia Tech, and a Ph.D. from the University of Arizona. His research interests relate to predictive analytics, with applications in online fraud and security, text mining, health, and social media. He has published more than 50 peer-reviewed articles in top journals and conference proceedings, including *MIS Quarterly, Journal of Management Information Systems, ACM Transactions on Information Systems, IEEE Transactions on Knowledge and Data Engineering*, and *IEEE Intelligent Systems*. His projects on Internet fraud, cyber security, and social media analytics have been funded by the National Science Foundation. He received the IBM Faculty Award and AWS Research Grant for his work on big data. He has also received best paper awards from *MIS Quarterly*, the *Association for Information Systems*, and the *Workshop on Information Technologies and Systems*. He serves as an associate editor for *Information Systems Research, Decision Sciences Journal, ACM Transactions on MIS*, and *IEEE Intelligent Systems*. He also serves on program committees for various conferences related to computational linguistics, text analytics, and data mining. His work has been featured in several media outlets, including the *Wall Street Journal,* the Associated Press, and Fox News.

FATEMEH “MARIAM” ZAHEDI is currently Roger L. Fitzsimonds Distinguished Scholar and was Wisconsin Distinguished Professor (1997–2007) in the information technology management area at the Sheldon B. Lubar School of Business, University of Wisconsin-Milwaukee. She has received her doctoral degree from Indiana University. Her present areas of research include web-based systems design and issues including security, privacy, trust, loyalty, personalized intelligent interface, health care, decision support systems, and web analytics for business. She has published more than 60 papers in major refereed journals, including: *MIS Quarterly, Information Systems Research, Management Science, Journal of Management Information Systems, Decision Support Systems, Information and Management, Decision Sciences, IEEE Transactions on Software Engineering*, and others. She has more than 50 publications in premier conference proceedings, and is the author of two books. She has served as a senior editor for *MIS Quarterly*, an associate editor of *Information Systems Research*, and has been on the editorial board of *Journal of Management Information Systems* and others. Her research has been funded by the National Science Foundation and other funding sources. She has received several awards for her teaching and research scholarship.

Journal of Management Information Systems / Spring 2015, Vol. 31, No. 4, pp. 109–157.

ISSN 0742–1222 (print) / ISSN 1557–928X (online)
DOI: 10.1080/07421222.2014.1001260

Daniel Zeng received M.S. and Ph.D. degrees in industrial administration from Carnegie Mellon University and a B.S. degree in economics and operations research from the University of Science and Technology of China, Hefei, China. He is a research fellow at the Institute of Automation, Chinese Academy of Sciences, and Gentile Family Professor in the Department of Management Information Systems at the University of Arizona. His research interests include intelligence and security informatics, infectious disease informatics, social computing, recommender systems, software agents, spatial-temporal data analysis, business analytics, and online advertising. He has published one monograph and more than 300 peer-reviewed articles. He serves as the editor in chief of *IEEE Intelligent Systems*. His research has been funded mainly by the U.S. National Science Foundation, the U.S. National Institutes of Health, the U.S. Department of Homeland Security, the National Natural Science Foundation of China, and the Ministry of Health of China. As principal investigator (PI) or co-PI, he has received more than $20 million in government research support. He is president elect of the IEEE Intelligent Transportation Systems Society and the past chair of INFORMS College on Artificial Intelligence.

Yan Chen is an assistant professor in the College of Business, Auburn University at Montgomery. She received her Ph.D. in MIS from the University of Wisconsin-Milwaukee. Her work has focused on information security, cyber espionage, privacy, and e-commerce. Her research has been published or accepted in journals including the *Journal of Management Information Systems* and the *Journal of Computer Information Systems*, and a number of refereed conference proceedings.

Hsinchun Chen is University of Arizona Regents Professor and Thomas R. Brown Chair in Management and Technology in the Management Information Systems (MIS) Department at the Eller College of Management. He joined the NSF as program director of the Smart and Connected Health Program in September 2014. He received a B.S. degree from National Chiao-Tung University in Taiwan, an MBA degree from SUNY Buffalo, and a Ph.D. degree in information systems from New York University. He is director of the Artificial Intelligence Lab where he developed the COPLINK system, which has been cited as a national model for public safety information sharing and analysis, and has been adopted in more than 3,500 law enforcement and intelligence agencies. He is a fellow of IEEE and AAAS and has received the IEEE Computer Society 2006 Technical Achievement Award, the 2008 INFORMS Design Science Award, and several other distinctions. He is author/editor of 20 books, 25 book chapters, 280 SCI journal articles, and 150 refereed conference articles covering digital library, data/text/web mining, business analytics, security informatics, and health informatics. His overall h-index is 70, with 17,000 citations. He is editor in chief (EIC) emeritus of the *ACM Transactions on Management Information Systems* (*ACM TMIS*) and EIC of *Security Informatics*. He has received over 90 grants totaling more than $40 million in research funding from NSF, NIH, NLM, DOD, DOJ, CIA, DHS, and other agencies.

Jay F. Nunamaker Jr. is Regents and Soldwedel Professor of MIS, Computer Science and Communication and director of the Center for the Management of Information and the National Center for Border Security and Immigration at the University of Arizona. He received his Ph.D. in operations research and systems engineering from Case Institute of Technology, an M.S. and a B.S. in engineering from the University of Pittsburgh, and a B.S. from Carnegie Mellon University. He received his professional engineer's license in 1965. He was inducted into the

Design Science Hall of Fame in May 2008 and received the LEO Award for Lifetime Achievement from the Association for Information Systems (AIS) in December 2002 and was elected a fellow of the AIS in 2000. He was featured in the July 1997 issue of *Forbes Magazine* on technology as one of eight key innovators in information technology. He is widely published, with an h-index above 60. His specialization is in the fields of system analysis and design, collaboration technology, and deception detection. The commercial product GroupSystems ThinkTank, based on his research, is often referred to as the gold standard for structured collaboration systems. He was a research assistant funded by the ISDOS project at the University of Michigan and an associate professor of computer science at Purdue University. He founded the MIS Department at the University of Arizona in 1974 and served as department head for 18 years.

ABSTRACT: Phishing websites continue to successfully exploit user vulnerabilities in household and enterprise settings. Existing anti-phishing tools lack the accuracy and generalizability needed to protect Internet users and organizations from the myriad of attacks encountered daily. Consequently, users often disregard these tools' warnings. In this study, using a design science approach, we propose a novel method for detecting phishing websites. By adopting a genre theoretic perspective, the proposed genre tree kernel method utilizes fraud cues that are associated with differences in purpose between legitimate and phishing websites, manifested through genre composition and design structure, resulting in enhanced anti-phishing capabilities. To evaluate the genre tree kernel method, a series of experiments were conducted on a testbed encompassing thousands of legitimate and phishing websites. The results revealed that the proposed method provided significantly better detection capabilities than state-of-the-art anti-phishing methods. An additional experiment demonstrated the effectiveness of the genre tree kernel technique in user settings; users utilizing the method were able to better identify and avoid phishing websites, and were consequently less likely to transact with them. Given the extensive monetary and social ramifications associated with phishing, the results have important implications for future anti-phishing strategies. More broadly, the results underscore the importance of considering intention/purpose as a critical dimension for automated credibility assessment: focusing not only on the "what" but rather on operationalizing the "why" into salient detection cues.

KEY WORDS AND PHRASES: design science, data mining, phishing websites, genre theory, Internet fraud, website genres, credibility assessment, phishing.

Phishing websites are fraudulent websites used to deceive unsuspecting Internet users [1]. Phishing websites have become increasingly pervasive, generating billions of dollars in fraudulent revenue [15, 29, 73]. In 2013, phishing attacks increased 87 percent relative to the previous year [39]. According to a Gartner report on industry adoption of security tools, increased phishing attacks have caused demand for web fraud detection software to reach all-time highs [21]. Phishing websites span numerous domains, including financial, medical, legal, retail, social networking, and search/portal websites, just to name a few. For instance, fake antivirus software websites were recently used to defraud 43 million users [71]. Similarly, a group of fraudsters in China developed 7 bogus military hospital websites used to defraud nearly 10,000 people [6]. In 2011, Google agreed to pay $500 million to settle a lawsuit involving the presence of fraudulent online pharmacies in their sponsored search results [12].

The authentic and legitimate appearance of phishing websites makes it difficult for users to identify them as fraudulent [15, 24, 30]. Consequently various anti-phishing toolbars and algorithms have been proposed. The most commonly used toolbars are the ones that come with web browsers such as Internet Explorer and Firefox. These toolbars "lookup" URLs against blacklists composed of URLs taken from online fraud prevention communities such as PhishTank.com [77]. The poor coverage/recall of these systems and lengthy manual URL verification times for blacklisting have prompted the development of content-based classification methods that use learning-based algorithms in conjunction with *fraud cues*: content elements that may serve as indicators of a website's lack of authenticity [3, 9, 46]. Content-based methods have yielded improvements in phishing website detection performance [2, 3, 56].

However, existing anti-phishing methods remain problematic. State-of-the-art content-based detection methods are mostly designed for specific industry sectors, or geared toward particular categories of phishing, thereby lacking the degree of generalizability needed to thwart the myriad phishing attacks encountered in enterprise or household settings [15, 20, 22]. Furthermore, content-based methods rely on thousands of fraud cues extracted from numerous pages within the website. This process, which can take several seconds to perform, has important performance and usability implications in real-time detection environments [3]. Furthermore, existing detection rates continue to hinder usability; users often disregard warnings due to lack of confidence in the tools' predictions [15, 74]. Collectively, these issues have inhibited the effectiveness of anti-phishing mechanisms.

The research objective of this study is to develop an anti-phishing method that is demonstratively more effective than prior efforts with respect to: (1) overall phishing website detection rates; (2) generalizability across various industries and categories of phishing attacks; and (3) accuracy and appropriateness of user security decisions when encountering phishing attacks. In order to achieve this objective, we adopted the design science paradigm to guide the development of the proposed IT artifact [27]; the genre tree kernel-based method for detecting phishing websites. Unlike prior content-based approaches, the proposed method does not rely on text- or image-based information derived from the website content (e.g., body text, URLs, source code, and images). Instead, the proposed approach leverages (1) website genre composition and (2) website design structure differences between legitimate and phishing websites. By adopting a genre theoretic perspective [76], the proposed method utilizes fraud cues that are associated with differences in *purpose* between legitimate and phishing websites, and the resulting operational practicalities. This *abstraction* away from specific text- and image-based patterns and toward emphasizing differences in business and operating models between legitimate and fraudulent websites, manifested through genre composition and design structure, facilitates enhanced detection of concocted and spoof websites.

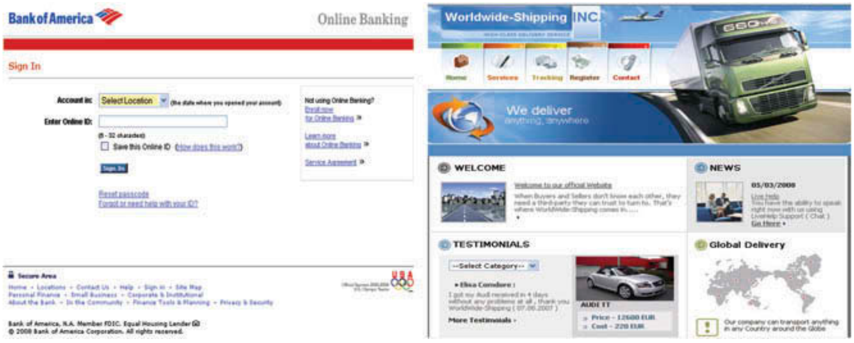
Experimental results on a large testbed revealed that the proposed approach outperformed existing state-of-the-art methods by a wide margin in a computationally efficient manner. Moreover, by using genre-based fraud cues instead of content-based cues commonly employed in prior approaches, the proposed method was more generalizable across various industry sectors, including retail, financial, legal, and medical websites. Furthermore, findings from a user study suggest that users utilizing the proposed method as compared to existing benchmark tools were able to better identify phishing websites, better avoid visiting phishing websites, and were also less likely to transact with phishing websites.

Related Work

In this section we provide relevant background on phishing website categories and also discuss the two prominent types of phishing website detection (i.e., anti-phishing) methods: lookup systems and content-based systems and methods [29].

Phishing Website Categories

The two most common categories of phishing websites are concocted and spoof sites [1]. Concocted websites attempt to appear as unique, legitimate commercial entities in order to engage in failure-to-ship fraud; accepting payment without providing the agreed upon goods/services [2, 3, 10]. Concocted websites often rely on social engineering-based attacks to reach their target audience. For instance, fraudulent eBay sellers may gain buyers' trust by going through a seller-controlled concocted online escrow website [1]. Concocted websites are becoming increasingly common, with hundreds of new entries added daily to online databases such as the Artists Against 419 [5]. In contrast, spoof websites engage in identity theft by mimicking legitimate websites and targeting those websites' customers, often through phishing emails [16, 20, 43, 46]. The Anti-Phishing Working Group has received reports of as many as 20,000 unique spoof URLs in a single month [3]. These spoof sites are used to attack millions of Internet users [9, 77]. Figure 1 shows examples of each



(the state where you opened your account)

(6 - 32 characters)

□ Save this Online ID (How does this work?)

Reset passcode
Forgot or need help with your ID?

Not using Online Banking?
Enroll now for Online Banking
Learn more about Online Banking
Service Agreement

Home · Locations · Contact Us · Help · Sign In · Site Map
Personal Finance · Small Business · Corporate & Institutional
About the Bank · In the Community · Finance Tools & Planning · Privacy & Security

Bank of America, N.A. Member FDIC. Equal Housing Lender
© 2008 Bank of America Corporation. All rights reserved.

Welcome to our official Website

When Buyers and Sellers don't know each other, they need a third-party they can trust to turn to. That's where WorldWide-Shipping comes in.....

05/03/2008
Live help
You have the ability to speak right now with us using LiveHelp Support (Chat)
Go there

• Elisa Condore :
I got my Audi received in 4 days without any problems at all , thank you WorldWide-Shipping (07.06.2007)
More Testimonials

Price - 12600 EUR
Cost - 220 EUR

Our company can transport anything in any Country around the Globe

Figure 1. Examples of Spoof (left) and Concocted (right) Websites

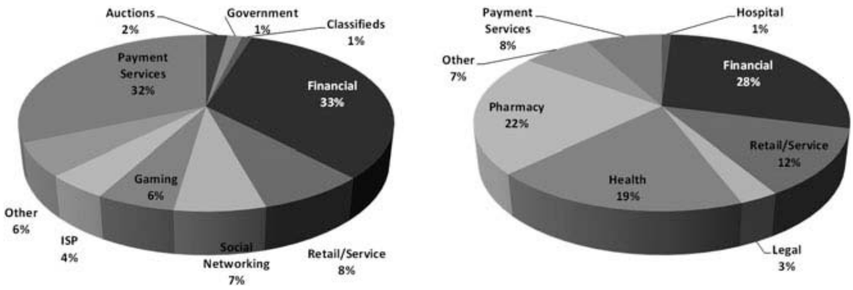


Figure 2. Proportion of Spoof (left) and Concocted (right) Websites by Industry Sector in 2012

category of phishing website; a spoof of Bank of America's website (left) and a concocted shipping company (right).

Phishing attacks target various industry sectors. According to data from the Anti-Phishing Working Group, Artists Against 419, and other fraud prevention communities, Figure 2 shows the proportion of spoof (left chart) and concocted (right chart) websites observed in 2012, by industry sector. From the left pie chart, it is evident that the most commonly spoofed categories are financial (including commercial and investment banks) and payment services (which includes PayPal, escrow service websites, etc.). Retail/service accounts for 8 percent of all spoofs, and includes e-tailers and shipping/courier/delivery service providers. Social networking websites such as Twitter, Facebook, Google +, MySpace, and LinkedIn are also increasingly spoofed for identity theft purposes. Auction websites such as eBay constituted 2 percent of unique spoofs in 2012. The "other" category, which includes search engines, universities, and various other industry sectors, accounted for 6 percent of total spoof attacks in 2012. Looking at the concocted websites' pie chart (right side of Figure 2), we see that while financial, retail/service, and payment services are still significant sectors, pharmacy and health websites are quite prevalent. Moreover, to a lesser extent, legal and hospital concocted websites are also represented. The figure underscores the extensiveness of phishing attacks across various industry sectors.

Lookup Systems

Lookup systems employ collaborative sanctioning mechanisms similar to those used for reputation ranking. They utilize a client-server architecture where the server side maintains a blacklist of known phishing URLs [77]. The blacklists are taken from online fraud prevention communities [10], such as PhishTank. Examples of lookup systems include Microsoft's IE Phishing Filter, Mozilla Firefox's FirePhish, Sitehound, Cloudmark, AntiPhish, and the GeoTrust TrustWatch toolbar [40, 74]. Given the popularity of the Microsoft IE and Firefox web browsers, these two browsers' associated security toolbars have the widest adoption [74, 77]. Prior studies have empirically demonstrated that lookup systems have very high precision

[77], but tend to also suffer from low recall levels, particularly on concocted websites [2]. Furthermore, phishing websites have a quick turnover rate: most have a life span of only a few days, just long enough to defraud a few users before being blacklisted [2, 16]. The situation is exacerbated by the fact that potential phishing URLs submitted require at least a couple of hours on average before they are verified in the blacklist [57]. By relying purely on reported URLs, blacklist-based approaches are constantly playing "catch-up," thereby making lookup systems susceptible to easy exploits [77].

Content-Based Tools and Methods

Content-based tools and methods detect phishing websites based on the appearance of fraud cues in website content and/or domain registration information. These fraud cues, which encompass image/visual similarity, body text phrases, URL and anchor text tokens, source code elements, and the quantity and types of hyperlinks, have yielded good results in prior studies [1, 3]. Examples of content-based tools include SpoofGuard, Netcraft, eBay's Account Guard toolbar, CANTINA, PhishDef, CANTINA+, Site-Watcher, and AZProtect [9, 2, 46, 78, 43, 75]. SpoofGuard uses image hashes, password encryption checks, URL similarities, and domain registration information [9]. Netcraft's classifier relies on domain registration information: domain name, host name, host country, and the registration date of the website [74]. EBay's Account Guard compares the visited website's content against that of eBay and PayPal's websites [77]. CANTINA uses the tf/idf (term frequency/inverse document frequency) from body text features [78]. CANTINA+ extends this feature set to also include PageRank scores, search results rankings, hash-based duplicate detection, login form detection, URL, and HTML features [75]. However, this method is mostly geared toward spoof websites; it is susceptible to concocted websites engaging in black-hat search engine optimization (as later demonstrated in the evaluation section). Similarly, PhishDef is geared toward identifying spoof websites that are buried deep on botnet-controlled hosts, resulting in long, randomly generated URLs [43]. Site-Watcher compares the visual and text similarity of the website of interest against a whitelist of known legitimate websites [46]. AZProtect uses thousands of body text, source code, URL and anchor text, linkage, and image-based attributes coupled with a linear kernel support vector machines (SVM) classifier [2]. Tools such as AZProtect and Netcraft have attained good results on both concocted and spoof websites, whereas SpoofGuard and Site-Watcher have yielded good performance on spoof websites [3, 46, 77]. In addition, several content-based anti-phishing algorithms have been proposed, though these are designed primarily to combat spoof websites (as opposed to concocted ones). Image- and URL-based features have been effectively exploited for spoof website detection [20, 43]. Other work has coupled domain registration and content-based features with an SVM classifier for spoof detection [56].

Shortcomings of Existing Methods

Generalizability: Prior content-based methods have typically "learned" fraud cues from training sets composed of legitimate and fraudulent websites associated specifically with either a certain industry sector (e.g., retail, financial, medical, etc.) or a specific category of phishing attack (e.g., spoof or concocted). Prior industry sectors that have received a fair amount of attention include financial, payment services, and medical websites [15, 22, 3]. Similarly, many prior studies have focused exclusively on spoof websites [20, 46]. These niche detectors lack generalizability. For instance, some spoof website detection tools rely heavily on URL-based features [43], which tend to be less effective by themselves when applied to concocted websites [3]. As another example, eBay's Account Guard tool is highly effective at identifying spoofed eBay and PayPal websites, but is unable to detect other categories of phish [77]. According to recent research on Internet browsing behavior, the average user spends at least 70 percent of total online time interacting with numerous retail, medical, news, social networking, search, and financial websites [23]. Hence, given the various industry sectors attacked by phishing websites (as previously shown in Figure 2), and Internet users' online behaviors, using a single niche detector or even several in concert is either ineffective or infeasible. There is a need for *more generalizable* methods for anti-phishing.

Lengthy run times unsuitable for real-time environments: Content-based methods that use text-, image-, and linkage-based attributes in conjunction with machine learning algorithms have attained good results for the specific domains or phishing categories investigated [3]. However, extracting thousands of attributes from hundreds of web pages per website can result in computational inefficiencies. Prior studies that used thousands of content-based features attained overall accuracies well over 90 percent for concocted and spoof detection, but with average classification times of 2.5 to 7 seconds per website [1, 3]. Given the small amount of time needed to inadvertently enter personal information in phishing websites, and usability concerns associated with lengthy tool response times, *computationally efficient* detection methods are highly desirable.

Ineffectiveness in user environments: Existing anti-phishing tools' performance has hindered their adoption and perceived usefulness; users are not very trusting of their recommendations [74, 41]. For instance, studies have shown that users employing popular browser-based toolbars frequently fall prey to phishing websites, with many attacks yielding success rates above 30 percent [74, 15]. Warning systems with high false positive/negative rates are susceptible to the "cry-wolf" effect, a behavioral response to the inadequate accuracy of a protective tool [18]. Improving phishing detection rates and generalizability in a computationally efficient manner should conceivably reduce attack success rates. However, given that users are often the weak link in the security loop [64], reducing the impact of phishing in both personal and organizational settings is largely dependent on individual web users' security behaviors [35, 41]. In summary, anti-phishing tools must be e*ffective in user environments* by facilitating enhanced security behavior.

Following the design science research paradigm [27], in the following section, we describe the proposed approach, which is designed to meet these objectives.

A Genre Theoretic Approach to Anti-Phishing

The design science paradigm provides guidelines for the development of IT artifacts, including constructs, models, methods, and instantiations [53, 54, 49, 27]. In design science, kernel theories are often used to govern the design process [67, 3]. Accordingly, in this study, we use genre theory to develop an anti-phishing method intended to provide enhanced detection capabilities, greater generalizability, and more effective phishing protection in user settings.

Genre theory states that genres are types of communication recognized and enacted by members of a community or organization [76]. Document genres combine purpose and form: the why and the what [55, 62]. Website genres include homepages, product/service pages, search pages, account/profile pages, frequently asked question (FAQ) sections, testimonials, newsletters, status and tracking pages, educational materials, publications, and so on [62, 66]. Each of these genres has a distinct and socially recognizable purpose [55]. For instance, testimonials are intended to increase credibility and consumer confidence whereas newsletters convey important information to an organization's employees and customers. Therefore, a website's genre composition is highly congruent with its overall objectives. Legitimate websites seek to increase visitors (new and repeat), conversions, and e-loyalty [14, 37]. Conversely, phishing websites are primarily focused on successfully defrauding unsuspecting users once (i.e., a hit-and-run approach). As a consequence of this "single conversion" objective, phishing websites often differ from legitimate ones in terms of their presence and frequency of certain website genres geared toward user experience and long-term loyalty [70]. For instance, prior analysis of hundreds of legitimate and phishing websites revealed that the phishing sites often failed to incorporate substantial FAQ sections or membership and login pages [1]. Conversely, phishing websites included an abundance of customer testimonials in order to gain Internet users' trust [3, 24].

Fraudsters' use of automated website development tools also results in structural design similarities between phishing sites [3, 70]. For instance, spoof sites often have more levels/depth than legitimate websites, as indicated by the number of slashes "/" in the URLS of these sites' web pages [3, 16]. In contrast, concocted websites tend to be relatively flatter, with web pages concentrated in a few levels [1]. Moreover, prior analysis has revealed that web pages at different levels also differ in terms of their quality, content, and genres [19]; with lower/bottom-level pages providing greater discriminatory potential for phishing website detection [1]. Information about a website's page levels can be derived from URL tokens and the file directory structure. The latter can also shed light on the location of key design-related files (e.g., images, banners, logos, scripts), which are often useful identifiers [16].

For complex structure information and problem-specific characteristics, kernel-based methods provide an effective alternative. Custom kernels have been used in recent document categorization and phishing website detection work [3, 45]. We propose a genre tree kernel that utilizes website genres applied to design structure information.

The Genre Tree Kernel for Phishing Website Detection

Kernel-based methods allow classifiers to integrate nonlinear information into a linear classifier using the *kernel trick*: the utilization of a kernel function that maps an input feature space into a high dimensional space without needing to know its explicit representation [13, 51]. Numerous types of kernels have been proposed in order to improve representational richness in various problem domains, including graph, tree, and string kernels [11, 48, 68]. Tree kernels have been used extensively for text mining and natural language processing [11, 44]. In this study, a genre tree kernel is proposed that creates a rooted tree from the website file directory structure, and labels the tree's file nodes with genre information. Details of the genre tree kernel are presented in the remainder of section.

Genre Tree Construction

Trees are constructed by traversing the websites' file directories (i.e., folders), beginning with the root directory. All files and folders contained in the root directory are considered its child nodes and are added to the tree with a label that corresponds to their file/folder name. Any child node folders (i.e., subfolders of the parent node) are also added to the traversal queue. The traversal and addition process is repeated until the contents of all subfolders have been added to the tree (i.e., until the queue is empty). Formally, the construction process results in a labeled rooted tree T with nodes $\{t_0 \ldots t_n\}$, where t_0 is the site's root directory and each node t_i has a label $v(t_i)$. We use $p(t_i) \in T$ to represent the parent node of t_i, whereas $c(t_i) \subset T \backslash \{t_0\}$ represents the set of children of the node t_i with cardinality $|\mathrm{c(ti)}|$ for all $i > 0$. It is important to note that for any file node t_i, $c(t_i) = \emptyset$, but a folder node may or may not have children. It is also worth noting that for most websites, the root folder name is the domain name. For websites with multiple domains, there may be multiple root directories. For these sites, the various root folders are placed under a single overarching folder that serves as an artificial root node.

Once the tree has been constructed the nodes are relabeled. Indexable file nodes are labeled with genre information. We utilized 18 website genres, most of which have been described in prior genre analysis studies [61, 62]. These are listed in Table 1. Prior website genre classification studies attained good results when using a page's URL tokens [47]. URL tokens have also been used in prior work on phishing website detection [43].

For a given indexable file node $t_i \in T \backslash \{t_0\}$, the genre classification is performed by analyzing $v(t_i)$ and $v(p(t_i))$; the node's filename and the node's parent folder's

Table 1. List of Website Genres Used to Construct the Genre Trees

Genre	Label	Description
About	A	Information about the organization, including history, background, and philosophy.
Career	E	Employment opportunities, pages with employee/workplace testimonials, and other pages with career information for potential hires.
Contact	C	Contact-related content, including locations, phone numbers, comment posting, e-mailing/speaking with representatives, e-mail sign-up, and live chat.
Event	V	Upcoming events, calendars, and lists of activities.
FAQ	Q	Frequently asked questions and other Q&A pages.
Homepage	H	The website's starting page.
Information	N	Articles, editorials, columns, and other informational resources.
Login	L	Login, logout, password retrieval, new member registration, and other account access related content.
Order	O	Order and shopping cart information, including order tracking and shipping.
Outreach	U	Community involvement, giving programs, scholarships, grant applications, and various other outreach-related activities.
Policy	P	Policies, terms, guarantees, and privacy notes.
Price	D	Fees, rates, prices, and quotes.
Product	R	Description of products, services, programs, classes, etc.
Publication	B	Magazines, newsletters, white papers, case studies, and other online publications.
Search	S	Search and navigation pages, including site maps and directories.
Social Media	M	Forums, blogs, and other social media content integrated into the website.
Support	W	Donations, volunteering, and other philanthropic opportunities.
Testimonial	T	Testimonials from customers, clients, students, patients, etc.

Note: Image and folder files are labeled with "I" and "F," respectively, while all remaining (unidentified) files are labeled with "X."

name. The two strings $v(p(t_i))$ and $v(t_i)$ are concatenated, tokenized, and stemmed using the porter stemmer [58]. The set of stemmed tokens is compared against learned feature sets of tokens associated with folder/file names belonging to the 18 aforementioned genres. Details regarding the genre feature set, and its effectiveness are presented in the ensuing paragraph. The indexable files' genres are assigned using a simple token-matching scheme, where they are categorized as belonging to the genre with the most matches. The formulation is presented in Figure 3. Image and folder files are relabeled with "I" and "F", respectively. All remaining files are relabeled with an "X" to indicate that they are unidentified.

The key word/token sets associated with each website genre were automatically learned from a training set composed of over 2,000 legitimate and phishing websites. We randomly extracted several hundred pages. Each page's website genre was

Let $A(t_i) = \{a_1 ... a_v\}$ denote the stemmed tokens taken from the concatenatation of $v(p(t_i))$ and $v(t_i)$, where $t_i \in T \setminus \{t_0\}$ is an indexable file node,

Let $B(j) = \{b_1 ... b_m\}$ denote the key word set associated with genre j, where $L(j)$ is the label associated with j, as described in Table 1.

The number of matches between t_i and j is computed as follows:

$$s(t_i, j) = \sum_{k=1}^{v} \sum_{p=1}^{m} x(a_k, b_p)$$

$$\text{where } x(a_k, b_p) = \begin{cases} 1, \text{if } a_k = b_p \\ 0, \text{if } a_k \neq b_p \end{cases}$$

And t_i is assigned to the genre with the greatest number of matches:

$$v(t_i) = \begin{cases} L\left(\underset{j}{\arg\max}\, s(t_i, j)\right), \text{if } \sum_{j=1}^{18} s(t_i, j) > 0 \\ \text{X}, \text{otherwise} \end{cases}$$

Figure 3. Genre Tree Indexable Node-Labeling Mechanism

tagged by an independent annotator. The information gain heuristic was used to learn the set of key words most indicative of a particular website genre. This was done by using a series of binary, one-against-all comparisons (one comparison for each genre in Table 1). Hence, the weight of a particular attribute b_x associated with genre j is based on its level of entropy reduction $H(Y) - H(Y|b_x)$, where $H(Y)$ is the entropy across the two classes ($Y = j$ or $Y \neq j$) in the training set, and $H(Y|b_x)$ is the entropy of $Y|b_x$. For each genre, the features with an information gain value above a certain threshold were incorporated. In order to assess the effectiveness of the genre tree node-labeling feature set and classification mechanism, we used cross-validation on the training pages and found that the method attained good precision rates with a very low run time. As later demonstrated in the evaluation section, the reduced computation time is highly beneficial: on average, the proposed genre node-labeling mechanism took less than 0.1 seconds per website. Table 2.

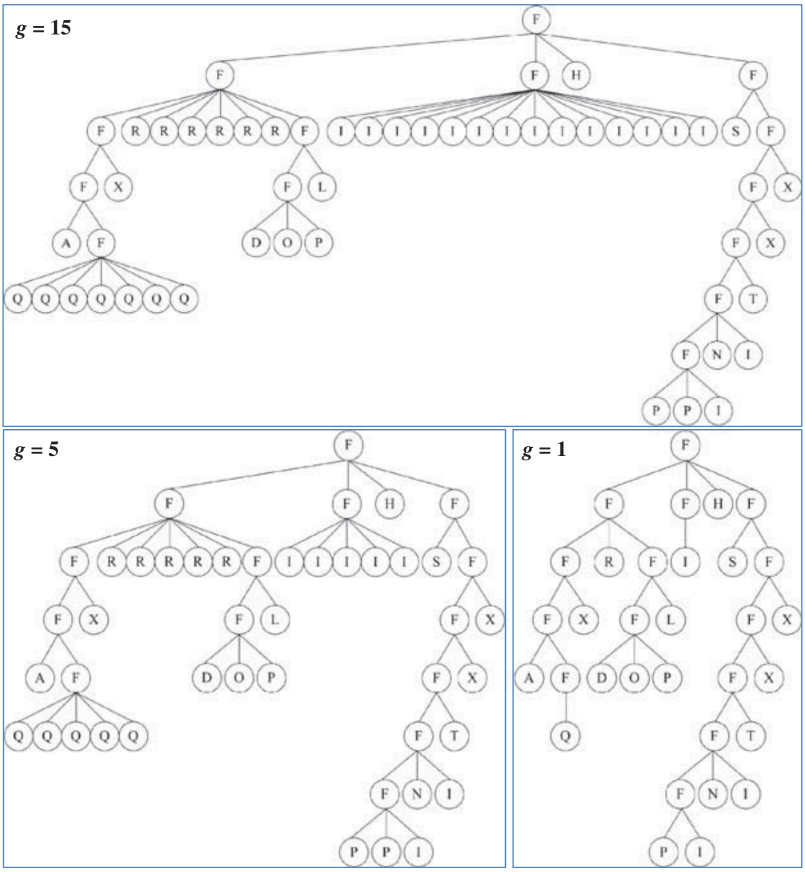
Websites often vary considerably in terms of their size (i.e., number of web pages). While such disparities in website sizes can signify important discriminators in some instances, they can also skew comparisons, often resulting in the inclusion of size-based biases. In order to improve the accuracy of comparisons, as well as computation times, website pages are often pruned [19, 42]. One common pruning strategy is

Table 2. Examples of Learned Features Used By Genre Node Labeling Mechanism

Genre	Examples of learned tokens
Login	Login, password, register, registration, signup, account, sign-in, enrollment
Information	Article, news, resource, info, information, facts, details,
Policy	Policy, privacy, term, guarantee, terms of use, legal, condition, promise
Price	Price, fee, payment, pay, quote, rate

to limit the maximum number of pages associated with a particular label. For instance, prior work on topic-based website categorization-pruned web pages containing duplicate topical information for enhanced performance [19]. Since certain website genres are more prevalent in terms of their occurrence frequency per website [47, 61, 62, 66], we use a genre-pruning parameter g to limit the number of sibling nodes associated with a particular genre. For a given node t_i, g indicates the maximum number of child nodes in $c(t_i)$ that can be labeled as belonging to genre j. Note that the g only limits the number of nonfolder child nodes. If the number of sibling nonfolder nodes sharing the same parent t_i exceeds g, some are randomly removed until the number in $c(t_i)$ belonging to genre j equals g.

Figure 4 shows an illustration of the impact of different values of g on the structure of a genre tree. For the same website, the figure shows genre trees constructed using g = 15, g = 5, and g = 1. Each node is labeled with a letter corresponding to one of the aforementioned genre categories, beginning with the root node, where $v(t_0) = F$. The g parameter impacts the structure and node composition of a genre tree, with smaller values of g resulting in narrower trees with a greater ratio of folder to file


g = 15

g = 5

g = 1

Figure 4. Impact of Different Values of g on the Structure of a Genre Tree

nodes. In this example, smaller values of g limit the number of image (I), product (R), FAQ (Q), and policy (P) sibling nodes.

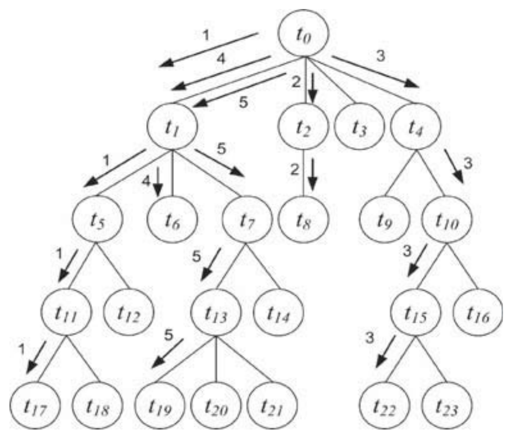
Genre Tree Traversal

Random walks provide a useful mechanism for traversing graph and tree structures. They have been used in prior work on graph kernels [38, 45]. The genre trees are traversed using a series of random walks. The w random walk paths are generated as follows. Beginning with the root directory node t_0, the random walk has a $(|c(t_0)| + 1)^{-1}$ probability of selecting any $t_i \in c(t_0)$ or terminating. In other words, if t_0 has three child nodes, they each have a ¼ probability of being selected, while the random walk termination probability is also ¼. If the walk is not terminated, from t_i, the random walk has a $(|c(t_i)| + 1)^{-1}$ probability of selecting any $t_k \in c(t_i)$ or terminating. Note that if $c(t_i) = \emptyset$, the probability of termination is 1. The random walk continues traversing the tree in a top-down manner until it is terminated. The process is repeated until w random walk paths have been generated. The formulation of the random walk traversal of the genre trees is presented in Figure 5.

Figure 6 shows an illustration of the random walk paths generated on a sample genre tree. The left-hand side shows the tree, along with w random path sequences, and the right-hand side lists the completed paths generated. The numbered arrows indicate subsequences associated with one of the w paths (here $w = 5$). All paths begin at t_0 and randomly traverse the tree nodes in a top-down manner until they either reach a childless node or are abruptly terminated (as described above). For instance, the first walk path begins at node t_0 and works its way down to nodes t_1, t_5, and t_{11}, before stopping at t_{17}.

```
Let $D(T) = \{q_1 \ldots q_w\}$ denote the set of random walk paths associated with genre tree $T$,
    where $q_x$ is a specific random walk path sequence that is generated as follows:
    Initially, $q_x = (t_0), t_i = t_0$, and $y = 1$,
    While $y = 1$
        Let $r = [0,1)$ represent a randomly generated number
        If $r < \frac{|c(t_i)|}{|c(t_i)|+1}$
            Select $t_k$, the $\left\| c(t_i)r + \frac{1}{2} \right\|$ element in $c(t_i)$
            $q_x = (q_x, t_k)$
            $t_i = t_k$
        Else
            $y = 0$
        End
    Loop
```

Figure 5. Genre Tree Traversal Formulation



1: $t_0 \rightarrow t_1 \rightarrow t_5 \rightarrow t_{11} \rightarrow t_{17}$
2: $t_0 \rightarrow t_2 \rightarrow t_8$
3: $t_0 \rightarrow t_4 \rightarrow t_{10} \rightarrow t_{15} \rightarrow t_{22}$
4: $t_0 \rightarrow t_1 \rightarrow t_6$
5: $t_0 \rightarrow t_1 \rightarrow t_7 \rightarrow t_{13} \rightarrow t_{19}$

Figure 6. Genre Tree Traversal Illustration

Genre Tree Comparison

The genre trees from any two websites are compared based on the path match lengths of their random walk paths. Figure 7 shows the formulation of the genre tree comparison used to generate the kernel matrix K. Given two genre trees T and T', each of the w random walk paths $\{q_1 \ldots q_w\}$ associated with T are compared against those belonging to T' (i.e., $\{r_1 \ldots r_w\}$), resulting in w^2 comparisons. Each path q_k from T is matched to a maximum of one identical path r_p from T', resulting in 0 to w total matches and a similarity score between 0 and 1 for each comparison $K(T, T')$.

It can be shown that the proposed kernel meets Mercer's conditions. For $K(T, T')$, where $T \neq T'$, since each path q_k in T can only be matched to a single path in T': $\sum_{p=1}^{w} L(q_k, r_p) M(q_k) M(r_p) \in \{0, 1\}$.

Let $\{q_1 \ldots q_w\}$ and $\{r_1 \ldots r_w\}$ represent the set of random walks along genre trees T and T'

$$K(T,T') = \sum_{k=1}^{w} \sum_{p=1}^{w} \frac{L(q_k, r_p) M(q_k) M(r_p)}{w}$$

where:

$$L(q_k, r_p) = \begin{cases} 1, & \text{if } \left(v(q_{k1}) \ldots v(q_{kh})\right) = \left(v(r_{p1}) \ldots v(r_{ph})\right) \\ 0, & \text{otherwise} \end{cases}$$

$$M(q_k) = \begin{cases} 1, & \text{if the path } q_k \text{ has not yet been matched to any of the paths of } T' \\ 0, & \text{otherwise} \end{cases}$$

w is the length of q, and the length of r.

Figure 7. Formulation of Genre Tree Comparison

Hence, $K(T,T') = \left[\sum_{k=1}^{w} \frac{0}{w}, \sum_{k=1}^{w} \frac{1}{w}\right] = \left[\frac{0}{w}, \frac{w}{w}\right] = [0,1]$. Moreover, since $L(q_k, r_p) = L(r_p, q_k)$, $K(T, T') = K(T', T)$. Finally, $\sum_{p=1}^{w} L(q_k, q_p)M(q_k)M(q_p) = 1$ since any path in T will match itself (i.e., when $k = p$). Hence, $K(T,T) = \sum_{k=1}^{w} \frac{1}{w} = \frac{w}{w} = 1$.

Evaluation

Leveraging existing URL databases maintained by online fraud prevention communities, we collected a testbed encompassing numerous concocted and spoof sites. Phishing websites typically have a short life span. In order to effectively collect them before they disappeared, we developed an automated spidering program. The collected web pages included complete body text, design code, URLs, images, and links. The training data set was composed of over 6,000 legitimate, concocted, and spoof websites. A separate testbed of 4,050 websites (1,350 legitimate, 1,350 concocted, and 1,350 spoof) was used for evaluation.

Table 3 The spoof website URLs were taken from two online repositories: Phishtank.com and the Anti-Phishing Working Group. The spoofs encompassed replicas of legitimate websites such as eBay, PayPal, Escrow.com, banks, university websites, search engines, social networking sites, and so on. The concocted website URLs were taken from online databases such as Artists Against 419, Escrow-fraud.com, LegitScript, and Health on the Net [4]. These included websites pertaining to shipping, financial, escrow, legal, and retail services as well as medical websites pertaining to hospitals, pharmacies, and health and wellness-related information. The 1,350 legitimate websites included ones that are commonly spoofed and also those

Table 3. Testbed Summary

Category	Industry sectors	Quantity	Sources
Legitimate sites	eBay, PayPal, Shipping, Financial, Escrow, Legal, Retail, University, Search Engine, Hospital, Pharmacy, Health, and so on	1,350	Commonly spoofed websites as well as types associated with the concocted sites.
Concocted sites	Shipping, Financial, Escrow, Legal, Retail, Hospital, Pharmacy, Health, and so on	1,350	Artists Against 419 (www.aa419.org) Escrow Fraud Prevention (escrow-fraud.com) LegitScript (www.legitscript.com) Health on the Net (www.hon.ch)
Spoof sites	Shipping, eBay, PayPal, Financial, Escrow, Retail, University, Search Engine, Social Networking, Pharmacy, and so on	1,350	PhishTank (www.phishtank.com) Anti-Phishing Working Group (www.antiphishing.org)

belonging to areas relevant to the concocted website testbed. Additional details regarding industry sectors in the testbed are presented in Appendix A.

Consistent with the design science research paradigm [27], we rigorously evaluated our information technology (IT) artifact. Four experiments were conducted to evaluate the genre tree's effectiveness with respect to the research gaps identified earlier. The evaluation metrics employed in the experiments included those used in prior research: overall accuracy and class-level precision, recall, and *F*-measure [3, 17, 70]. Details regarding these metrics appear in Appendix B. A brief overview of the experiments and associated research questions is as follows. In Experiment 1, we compared the genre tree kernel against various website content-based kernel and non-kernel classification methods used in prior research, and investigated the following research questions:

> *RQ1a: How effective is the genre tree kernel, versus benchmark content-based methods, in terms of legitimate, concocted, and spoof recall rates?*

> *RQ1b: Can the genre tree kernel attain significant performance gains over benchmark content-based methods across various industry sectors and phishing website categories?*

Experiment 2 compared the genre tree kernel against existing phishing website detection tools/systems to shed light on the following research questions:

> *RQ2a: How effective is the genre tree kernel compared to benchmark anti-phishing tools in terms of legitimate, concocted, and spoof recall rates?*

> *RQ2b: Can the genre tree kernel attain significant performance gains over benchmark anti-phishing tools across various industry sectors and phishing website categories?*

> *RQ2c: How do the genre tree kernel's classification times compare against the top-performing anti-phishing tools?*

In Experiment 3, the genre tree kernel was compared against various alternative methods that incorporate genre and/or tree information. The key research question is as follows:

> *RQ3: How does the genre tree kernel's legitimate, concocted, and spoof detection rates compare against alternate genre- and tree-based kernel methods?*

In Experiment 4, a user study was conducted comparing the genre tree kernel against two existing benchmark tools in terms of users' ability to identify and avoid phishing websites.

> *RQ4a: Will users utilizing the genre tree kernel identify phishing websites more effectively, in terms of legitimate and phishing recall, than those employing benchmark tools?*

> *RQ4b: Will users utilizing the genre tree kernel avoid visiting phishing websites more effectively than those employing benchmark tools?*

RQ4c: Will users utilizing the genre tree kernel be less willing to transact with phishing websites as compared to those employing benchmark tools?

Collectively, Experiments 1–3 were intended to demonstrate the overall phishing detection effectiveness, generalizability, and enhanced run times for the genre tree kernel, whereas Experiment 4 was intended to evaluate the genre tree kernel's ability to improve users' security decision making when encountering phishing attacks.

Experiment 1: Comparison with Content-Based Classifiers

Prior phishing website detection work has relied on content-based features such as web page text, code, images, URLs, and link information, in conjunction with existing kernel and non-kernel classification methods. Extended feature sets encompassing an amalgamation of features from across these categories have been empirically shown to provide the best performance, outperforming individual categories [1, 3]. Accordingly, we constructed a rich feature set composed of over 9,000 attributes derived from the websites' body text, source code, URL tokens, images, and linkage-based information [2, 17, 70]. These features were learned from the training data set, using the information gain heuristic. The web page body text attributes encompassed over 4,500 word phrases, lexical measures, and spelling mistakes [3]. The URL fraud cues were 2,500 words and characters derived from the URL and anchor text [43]. Source code fraud cues included 1,000 items pertaining to code commands as well as general programming style markers [2, 70]. The image features included pixel color frequencies arranged into 1,000 bins as well as 40 image structure attributes, including image height, width, file extension, and file size [20, 46]. These attributes were intended to detect the presence of duplicate/recurring images in concocted and spoof sites [9]. The linkage features were composed of 50 attributes related to the number of incoming/outgoing links at the site and page levels [2].

We compared the proposed genre tree kernel against various kernel and non-kernel methods, all of which were run using the extended content-based feature set. The comparison kernels included the linear composite kernel proposed by [3], as well as the standard linear, radial basis function (RBF), and polynomial kernels, all of which have worked well in prior studies on concocted and spoof sites [3, 17, 56]. The kernels were run using the Support Vector Machines (SVM) classifier in the SVM Light package [34]. We also evaluated several non–kernel-based classification methods used in related prior work, including logistic regression, Bayesian network, J48 decision tree, neural network, and naive Bayes [1, 3, 52]. These classification methods were run using WEKA [72]. The GT kernel's w and g parameters were tuned using 10-fold cross-validation on the training data, and were set to $w = 30$ and $g = 10$. All comparison methods underwent extensive parameter tuning to ensure the best possible results.

Table 4 shows the experimental results. The genre tree kernel significantly outperformed all comparison methods (kernel- and non–kernel-based) in terms of legitimate, concocted, and spoof detection rates. All paired t-test p-values were

Table 4. Experiment Results for Genre Tree Kernel and Comparison Content-based Methods

		Legitimate websites (n = 1,350)			Concocted detection (n = 1,350)			Spoof detection (n = 1,350)		
Learning technique	Overall accuracy (n = 4,050)	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.
Genre tree*	97.01	95.59	94.18	97.04	95.65	96.96	94.37	98.35	97.11	99.63
Linear composite	92.15	88.76	84.91	92.96	89.48	92.49	86.67	94.99	93.22	96.81
Linear	90.79	86.83	82.94	91.11	87.83	90.56	85.26	93.71	91.53	96.00
Polynomial 2nd	90.00	85.72	81.78	90.07	86.98	89.50	84.59	92.89	90.57	95.33
RBF	89.46	85.03	80.71	89.85	86.62	89.24	84.15	92.29	90.29	94.37
Polynomial 3rd	88.72	84.07	79.39	89.33	85.91	88.65	83.33	91.58	89.76	93.48
Logit regression	88.35	83.57	78.84	88.89	86.49	88.40	84.67	90.31	89.17	91.48
J48 decision tree	87.56	80.81	83.15	78.59	84.21	80.49	88.30	88.20	81.73	95.78
Bayesian network	87.14	81.80	77.40	86.74	86.15	86.60	85.70	87.99	87.03	88.96
Naive Bayes	81.68	75.58	68.01	85.04	83.18	84.54	81.85	80.94	83.93	78.15
Neural network	75.06	70.28	58.30	88.44	74.50	85.14	66.22	77.46	85.92	70.52

* Significantly outperformed all comparison methods (paired t-test p-values appear in Appendix C, Table C1).

less than 0.001 (see Appendix C, Table C1 for details). The performance gain in terms of overall accuracy was approximately 5 percent over the best kernel method (linear composite). Moreover, the genre tree kernel outperformed the best non-kernel method, logit, by over 8 percent with respect to overall accuracy. An important factor contributing to the genre tree kernel's enhanced performance over kernel-based methods was its ability to better detect concocted websites; it outperformed comparison kernel-based techniques by at least 7 percent in terms of concocted recall. It also improved legitimate and spoof detection rates by at least 3 percent (based on recall values).

With respect to the comparison methods, the kernel-based techniques outperformed the non–kernel-based ones in terms of overall accuracy. Interestingly, the performance of certain non-kernel methods such as the J48 decision tree was relatively decent in terms of concocted and spoof website recall, as compared to the kernel methods; however it lagged behind in terms of legitimate website recall. Consistent with prior work, the concocted website detection rates were lower, since this is considered a more challenging task as compared to spoof detection [3].

Figure 8 shows the performance of the genre tree kernel and four comparison methods on legitimate, concocted, and spoof websites, grouped by industry sectors. The three bar charts depict recall rates for industry sectors encompassing at least 25 instances in the test bed for that particular type of website (i.e., legit, concocted, spoof). The four comparison methods were the top-performing kernel-based

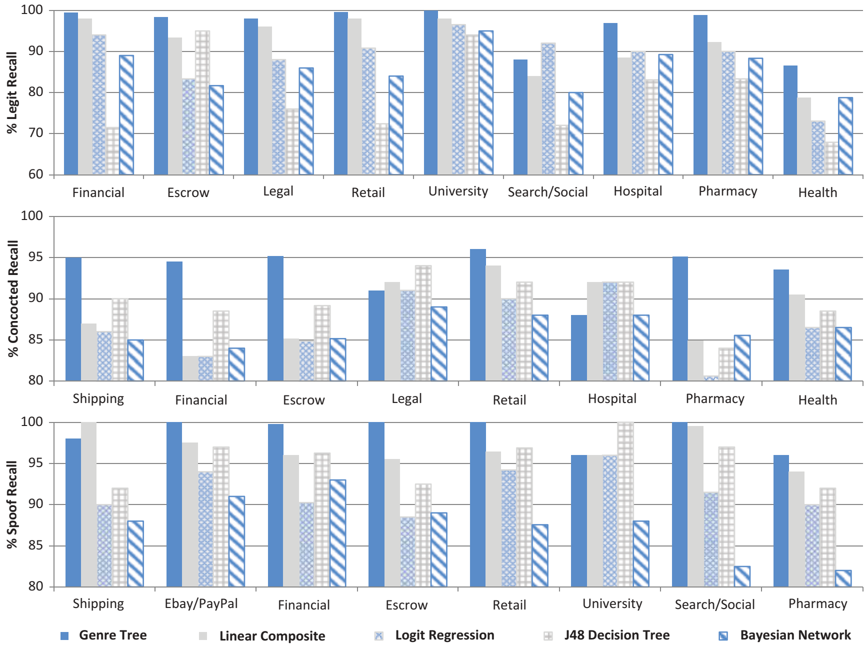


Figure 8. Performance of Genre Tree Kernel and Comparison Methods on Legitimate (top), Concocted (middle), and Spoof (bottom) websites, Grouped by Industry Sectors

technique (linear composite), as well as the top three non-kernel methods. Since other techniques were either correlated/redundant with and/or underperformed the chosen four, they were excluded from the figure.

Based on the top chart in Figure 8, it is evident that the genre tree kernel had the highest recall rates on legitimate websites for most industry sectors, including financial, escrow, legal, retail, university, hospital, pharmacy, and health. The one exception was search engines/social networking websites, where logistic regression performed better. Similarly, genre tree kernel outperformed the top comparison methods on most sectors of concocted and spoof websites (as shown in the middle and bottom charts). Overall, the genre tree kernel outperformed all comparison methods on 20 out of 25 industry sectors; it significantly outperformed the best comparison method (linear composite) on 16 cases (see Appendix C, Table C2). In contrast, performance for comparison methods varied. While the linear composite kernel performed well on various legitimate categories, it was outperformed by either logit regression, decision tree, or Bayesian network on escrow, search/social, hospital, and health websites. Similarly, the decision tree had better performance than the linear composite kernel on many concocted and spoof website industry sectors. By illustrating the consistently effective performance of the genre tree kernel, and inconsistent performance of even the best comparison methods, the results underscore the enhanced generalizability of the genre tree kernel across categories of phishing websites and industry sectors.

Experiment 2: Comparison with Existing Detection Tools

We evaluated the genre tree kernel in comparison with existing phishing website detection tools. The comparison tools were twelve systems that had performed well in prior testing and/or were commonly used [2, 77, 75]. Seven of the comparison tools were classifier systems (AZProtect, SpoofGuard, Netcraft, CANTINA, PhishDef, and CANTINA+, and eBay's Account Guard), four were lookup systems (IE Phishing Filter, FirePhish, EarthLink Toolbar, and Sitehound), and one was a hybrid tool that coupled content-based signatures with a lookup approach based on community feedback (Norton Safe Web). The lookup systems all utilized server-side blacklists that were updated regularly by the system providers. Five of the classifier systems, AZProtect, CANTINA, PhishDef, CANTINA+, and SpoofGuard, required training [3, 78, 43, 75, 9]. These five tools were also trained on the same 6,000 websites used by the genre tree kernel. All systems classified each of the 4,050 testbed websites as legitimate or phish, while the sites were still online. The results are presented in Table 5.

The genre tree kernel significantly outperformed all nine comparison tools in terms of overall accuracy and class-level *f*-measure, precision, and recall for real, concocted, and spoof websites. All paired *t*-test *p*-values were less than 0.01. The performance gain in terms of overall accuracy was 4 percent over the best existing tool (AZProtect). Once again, the genre tree kernel's ability to better detect

Table 5. Experiment Results for Genre Tree Kernel and Existing Tools

Learning technique	Overall accuracy (n = 4,050)	Legitimate websites (n = 1,350)			Concocted detection (n = 1,350)			Spoof detection (n = 1,350)		
		F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.
Genre tree*	97.01	95.59	94.18	97.04	95.65	96.96	94.37	98.35	97.11	99.63
AZProtect	92.62	89.43	85.53	93.70	90.04	93.25	87.04	95.48	93.91	97.11
CANTINA+	89.98	86.64	77.96	97.48	83.50	96.69	73.48	98.24	97.52	98.96
Netcraft	86.02	82.55	70.70	99.19	83.63	98.89	72.44	92.33	99.07	86.44
Norton Safe Web	83.41	80.07	66.77	100.00	78.82	100.00	65.04	92.00	100.00	85.19
PhishDef	80.02	76.46	62.90	97.48	62.02	94.54	46.15	96.88	97.31	96.44
CANTINA	79.33	76.20	61.84	99.26	72.36	98.72	57.11	89.52	99.10	81.63
SpoofGuard	71.80	68.10	54.66	90.30	63.31	83.97	50.81	80.76	88.45	74.30
eBay AG	39.65	52.49	35.58	100.00	2.63	100.00	1.33	29.97	100.00	17.63
IEFilter	62.42	63.95	47.01	100.00	23.30	100.00	13.19	85.11	100.00	74.07
Mozilla FirePhish	61.14	63.17	46.17	100.00	21.78	100.00	12.22	83.17	100.00	71.19
Sitehound	56.25	60.38	43.24	100.00	49.41	100.00	32.81	52.86	100.00	35.93
EarthLink	51.85	57.90	40.86	99.33	17.46	93.53	9.63	63.28	98.59	46.59

* Significantly outperformed all comparison methods (paired t-test p-values appear in Appendix C, Table C3).

concocted websites set it apart from the best comparison classifier-based detection tools. With respect to the comparison tools, AZProtect, CANTINA+, Netcraft, Norton Safe Web, and PhishDef had the best overall accuracies. Whereas tools such as CANTINA+ and PhishDef were effective against spoofs, they failed to accurately detect concocted websites, thereby making them less generalizable. As expected, the lookup systems had higher precision on the phishing websites, at the expense of considerably lower phishing recall. As with the previous experiment, nearly all tools had better performance on the spoof websites as compared to the concocted ones. The lookup systems' recall rates were particularly low on the concocted websites; popular security toolbars such as IE Filter and FirePhish detected less than 15 percent of concocted websites.

In addition to improved legit, concocted, and spoof recall rates, and better generalizability across website brands/categories, as evidenced by Tables 4 and 5 and Figure 8, the genre tree kernel also produced shorter run times than the best-performing content-based method: AZProtect. Across the testbed, the genre tree kernel had an average run time of 2.2 seconds (standard deviation of approximately 1 second), while AZProtect had an average run time of about 4 seconds with a standard deviation of over 2 seconds. Figure 9 illustrates the enhanced computation times for the genre tree kernel (left charts) as compared to AZProtect (right charts). The figure shows the run times on each concocted and spoof website in the testbed (*x*-axis), as well as the number of text and image files examined by AZProtect; due to computational constraints, the system only analyzes up to 50 web pages per website [2]. All websites are denoted by their industry sector (e.g., financial, escrow, retail, etc.).

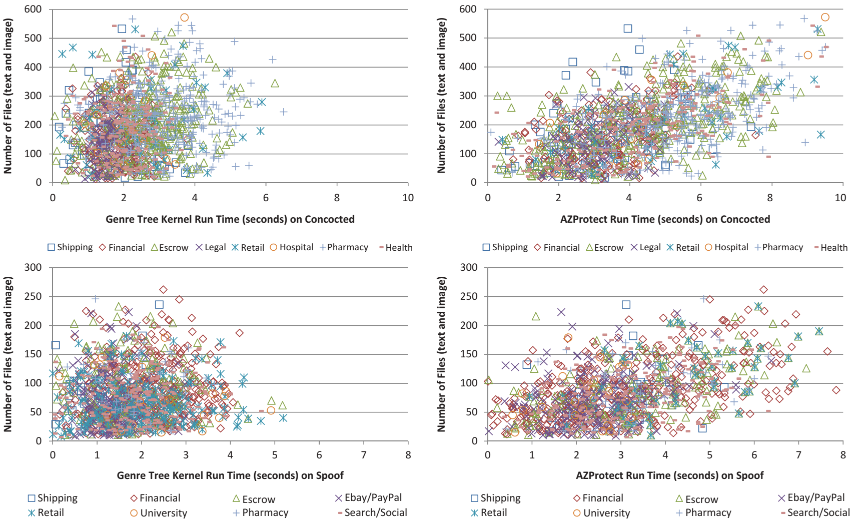


Figure 9. Run Times for Genre Tree and AZProtect on Concocted (top) and Spoof (bottom) Websites

From the figure, it is apparent that due to its extensive use of text- and image-based features, AZProtect's run times for concocted and spoof websites were correlated with the number of files examined. Hence, concocted retail, escrow, and pharmacy run times were highest for AZProtect, often 6–8 seconds. Overall, the genre tree kernel had significantly lower run times on 15 of 16 phishing type/industry sector combinations (see Appendix C, Table C5). These run times are highly problematic in real-time environments where users might inadvertently provide personal information to a phishing website within a few seconds. Furthermore, additional analysis revealed that the performance of the genre tree kernel was fairly robust for various parameter settings, as shown in Appendix D.

Experiment 3: Comparison with Alternate Genre and Tree Kernel Methods

The previous two experiments demonstrated the effectiveness of using the genre tree kernel over kernel and non-kernel content-based classification methods, as well as existing detection tools. These experiments shed light on the overall effectiveness of the proposed kernel. In this section, we explore the utility of the genre tree kernel's main components: (1) genre information; (2) tree structure; (3) random walk tree traversal; (4) genre tree comparison mechanism. In order to assess each component's contribution to the overall performance of the genre tree kernel, we evaluated various alternatives for each component. Specifically, four types of ablation settings were incorporated: genre trees using random walk paths (GT-RW), genre trees using other tree kernels (GT), kernels using genre presence vectors (G), and kernels applied to trees devoid of genre labels (T and T-RW). For all techniques, parameter tuning was performed on the 2,000 training website data set. Our method, labeled GT-RW-PM, was once again run using $w = 30$ and $g = 10$. The resulting parameter settings for each ablation setting/technique are included below, following the techniques' descriptions.

Genre trees using random walk paths (GT-RW): The genre-tree comparison mechanism was compared against three alternate matching approaches, all of which were also applied to the trees' random walk path sequences: cross entropy, contiguous path, and noncontiguous path. Given the strong relationship between prior research on path and sequence kernels and string matching [68], the comparison matching approaches used were based on techniques related to the latter.

Cross entropy uses the match length $L(x, y)$ between path sequences x and y as an indicator of the degree of similarity, where x and y are the concatenations of the w random walk paths [36]. $L(x, y)$ can be computed as follows: let $(x_c \ldots x_{c+h})$ and $(y_d \ldots y_{d+h})$ represent the largest common path subsequence between $(x_0 \ldots x_a)$ and $(y_0 \ldots y_b)$, where $(v(x_c) \ldots v(x_{c+h})) = (v(y_d) \ldots v(y_{d+h}))$. $L(x, y) = max(a, b)^{-1}h$. Cross entropy was run using $w = 30$ and $g = 5$.

Contiguous path is an adaptation of the n-gram kernel [63]. It projects all aggregations of the w random walk paths into a feature space indexed by all k-tuples of genre labels for some fixed k. The strength of the feature indexed by the

k-tuple $f = (f_1 \dots f_k)$ for an aggregated random walk path sequence of length d, is the frequency of all contiguous occurrences of f as a subsequence in d. The similarity $C(x, y)$ between path sequences x and y can be computed as $K(x, y)(K(x, x)K(y, y))^{-1/2}$, where $K(x, y)$ is the dot product of the two k-tuple feature vectors associated with x and y. Contiguous path was run using $k = 2$, $w = 30$, and $g = 10$.

Noncontiguous path is inspired by the string kernel [48]. Adapting the description from [63], it projects all aggregated random walk paths into a feature space indexed by all k-tuples of genre labels for some fixed k. The strength of the feature indexed by the k-tuple $f = (f_1 \dots f_k)$ for a random walk path of length d, is the sum over all contiguous or noncontiguous occurrences of f as a subsequence in d. Each occurrence of f is weighted by an exponentially decaying function of its length in d. The similarity $N(x, y)$ between path sequences x and y can be computed as $K(x, y)(K(x, x)K(y, y))^{-1/2}$, where $K(x, y)$ is the dot product of the two k-tuple feature vectors associated with x and y, respectively. Noncontiguous path was run with $k = 2$, $w = 30$, and $g = 10$.

Genre trees using alternative tree kernels (GT): In order to assess the efficacy of using random walk paths, we also evaluated two alternate genre tree kernels. Let $h_i(x)$ denote the presence of the ith tree fragment in website x (where $h_i(x) = 1$ if the ith tree fragment exists in x) such that x is now represented as a binary vector $h(x) = (h_1(x), h_2(x), \dots, h_n(x))$. The standard tree kernel $T(x, y)$ between websites x and y can be computed as [11]: $K(x, y)(K(x, x)K(y, y))^{-1/2}$, where $K(x, y)$ is the dot product of $h(x)$ and $h(y)$ multiplied by a decay parameter that is inversely proportional to the number of nodes in i. The kernel was run using $g = 1$.

While the standard tree kernel is based on the number of matching subtrees, the maximum subtree-based matching mechanism computes the largest common subtree in x and y, $L(h(x), h(y))$, in terms of number of nodes. $K(x, y)$ is then computed as $2L(h(x), h(y))(N_x + N_y)^{-1}$, where N_x represents the number of nodes in x. Normalization is performed, similar to the standard tree kernel. The kernel was run using $g = 5$.

Genre presence vectors (G): For each website, we derived a presence vector for the occurrence of the 21 genre labels (i.e., F, I, X, and the 18 mentioned in Table 1), each across levels 1–k of the website, where the root folder was considered level 0. This resulted is a vector $x = (x_1 \dots x_{21k})$ composed of $21k$ binary values (i.e., 21 genres multiplied by k levels). For instance, x_4 and x_5 represented the presence (or absence) of the About (A) and Contact (C) genres at level 1, respectively. The presences vectors were derived using $k = 10$, and were input into four kernels: linear, second- and third-degree polynomial, and RBF.

Trees devoid of genre labels (T and T-RW): We constructed labeled trees that did not contain genre information. All indexable file nodes were assigned a label of "T." Hence, the trees were composed of nodes with the following four labels: "F", "I", "X", and "T". All three random walk-based comparison methods were used. In other words, the three T-RW methods were cross entropy, contiguous path, and noncontiguous path. These were all run using $w = 30$ and $g = 10$. Alternate tree kernels (T) as well as number of subtrees and maximum subtree, were also evaluated. These were both run using $g = 5$.

Table 6 The first row depicts the proposed kernel, which combines a genre tree with a random walk tree traversal and exact pattern matching (GT-RW-PM). The

Table 6. Experiment Results for Genre Tree Kernel and Alternate Genre-based and Kernel Methods

Learning technique		Overall Accuracy (n = 4,050)	Legitimate websites (n = 1,350)			Concocted detection (n = 1,350)			Spoof detection (n = 1,350		
			F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.
	GT-RW-PM*	97.01	95.59	94.18	97.04	95.65	96.96	94.37	98.35	97.11	99.63
GT-RW	Cross Entropy	93.58	90.74	87.38	94.37	92.42	94.16	90.74	95.03	94.44	95.63
	Con Path	86.17	81.69	73.13	92.52	85.79	91.52	80.74	88.47	91.93	85.26
	Non-Con Path	81.26	76.23	66.03	90.15	80.21	88.19	73.56	84.32	89.04	80.07
GT	Max Subtree	93.14	90.25	85.73	95.26	93.00	95.05	91.04	94.12	95.16	93.11
	Num Subtree	90.15	86.62	79.15	95.63	90.44	95.17	86.15	91.86	95.30	88.67
G	Poly 3	92.05	88.32	86.56	90.15	90.07	90.13	90.00	93.27	90.69	96.00
	Poly 2	91.41	87.45	85.18	89.85	89.52	89.79	89.26	92.67	90.36	95.11
	RBF	91.41	87.39	85.53	89.33	89.05	89.28	88.81	92.94	90.01	96.07
	Linear	90.49	86.09	84.05	88.22	88.14	88.20	88.07	91.98	88.99	95.19
T and T-RW	Cross Entropy	91.80	88.55	82.84	95.11	92.11	94.82	89.56	92.77	94.89	90.74
	Max Subtree	91.16	87.88	80.92	96.15	90.43	95.70	85.70	93.75	95.97	91.63
	Num Subtree	89.41	85.18	79.81	91.33	86.46	90.52	82.74	92.84	91.57	94.15
	Con Path	81.16	74.91	67.36	84.37	80.29	83.23	77.56	82.72	83.92	81.56
	Non-Con Path	71.60	63.26	55.62	73.33	71.67	72.62	70.74	71.67	72.62	70.74

* Significantly outperformed all comparison methods (paired t-test p-values appear in Appendix C, Table C6).

comparison ablation methods were grouped into the four aforementioned categories: GT-RW, GT, G, and T and T-RW. Based on the experimental results, GT-RW-PM outperformed all comparison ablation settings by a fairly wide margin, garnering at least a 3 percent gain in overall accuracy over the next best setting. The performance gain was balanced across legitimate and phishing websites; GT-RW-PM had the best performance for all 10 evaluation metrics (with all paired t-test p-values below 0.01).

With respect to the comparison ablation settings, the genre tree-based methods GT-RW-CrossEntropy and GT-MaxSubTree had the best overall accuracies and legitimate website recall, with values of over 93 percent and 94 percent for these two metrics, respectively. The performance of the GT-RW methods was quite sensitive to the specific comparison method employed. While GT-RW-PM and GT-RW-CrossEntropy each attained very good results, conversely GT-RW-NonConPath was the only ablation setting using genre information that had accuracy below 85 percent.

The genre presence vector-based classifiers (e.g., G-Poly3, G-Poly2) had the most consistent performance, with overall accuracies in the 90–92 percent range for all four settings evaluated. Interestingly, the results using the genre presence vectors were on par with those attained using the best content-based feature vectors (see Table 4 presented earlier). The results underscore the discriminatory potential of genre information for anti-phishing. These findings suggest that when detecting phishing websites, analysis of website genre composition is equally as important as evaluation of website content.

The trees devoid of genre information (T and T-RW techniques) had lower accuracies as compared to their genre tree counterparts. However, a couple of these techniques, T-MaxSubTree and T-RW-CrossEntropy, also had overall accuracies greater than 90 percent. These methods outperformed all of the non-kernel content-based methods (as well as many of the kernel-based ones) appearing earlier in Table 4. The results further illustrate the utility of design structure information such as website depth/levels, as alluded to by prior research [1, 16].

The results of this experiment provide insights into how various components of the genre tree kernel contributed to its overall performance. For instance, using tree structure information alone was not as useful as utilizing genre information at different levels (see also Appendix E). Moreover, combining the two types of information increased the potential for enhanced overall accuracies, as illustrated by the best GT and GT-RW settings. In particular, the use of genre trees improved/reduced false positive rates over G. Legitimate recall values for the best GT-based techniques (i.e., GT-MaxSubtree, GT-RW-CrossEntropy) were 4–6 percent higher than those associated with the G settings. Conversely, these same GT-based methods had legitimate recall values comparable to the best T and T-RW settings (T-MaxSubTree and T-RW-CrossEntropy), but attained higher recall rates on both concocted and spoof websites.

These results suggest that the genre tree kernel was able to exploit the complementary information provided by a website's tree structure and genre composition. Moreover, the exact matching-based comparison mechanism utilized by the genre tree kernel yielded better results than alternate random walk measures (e.g., cross entropy, contiguous/noncontiguous path) and subtree-based measures (e.g., number of subtrees and max subtree). Collectively, these factors all contributed to the genre tree kernel's enhanced phishing website detection capabilities.

Experiment 4: User Study Evaluating Genre Tree Kernel

A controlled study was conducted to evaluate the utility of the genre tree kernel with users. In the experiment, users were each given a list of 10 URLs for bank websites and were asked two questions per URL: (1) whether they considered the website legitimate; and (2) whether they would consider opening a savings account from the website. The choice of website category and task were motivated by the fact that financial institution websites are among the most common categories for phishing attacks [60]. According to industry research, nearly 0.5 percent of banks' customers fall prey to phishing attacks annually [59]. Spoof and concocted online banks are highly successful at luring victims using the ruse of offering attractive banking services, currency exchange programs, small business loans, philanthropic ventures, and so on. [2].

The 10 URLs were displayed in a manner visually analogous to search engine results. This display method was considered appropriate because Internet users spend a considerable amount of time using search engines [23], and because search engines have recently been exploited considerably by phishing and other types of illicit websites [25, 7, 65]. Five of the 10 URLs displayed were for legitimate banks while the other 5 were for phishing bank websites. Both types of phishing websites were incorporated and half of the users were assigned to either type. Hence, half were given 5 concocted banks and 5 legitimate bank website URLs, while the other half were provided 5 spoof banks and 5 legitimate bank website URLs. The 10 URLs provided to each user were displayed in random order.

The 5 legitimate and 5 concocted/spoof URLs displayed to each user were randomly selected from a pool of 45 websites taken from our test bed (15 legitimate, 15 concocted, and 15 spoof). The 15 legitimate websites in the pool were roughly balanced based on the total size of the banks, using deposit volume data provided by the Federal Deposit Insurance Corporation (i.e., with some very large, some medium, and some small banks). The 15 spoof websites utilized in the pool were replicas of the 15 legitimate websites. The 15 concocted websites employed in the pool were also taken from the testbed.

An anti-phishing tool was used to provide warnings. Each participant was randomly given one of three tools: the genre tree kernel, AZProtect, and IE Phishing Filter. This resulted in a 2 × 3 factorial design with six total experiment settings (i.e., spoof-genre tree, spoof-AZProtect, spoof-IEFilter, concocted-genre tree, concocted-AZProtect, and concocted-IEFilter). AZProtect was adopted since it represents a state-of-the-art content-based method. IE Phishing Filter was employed since its performance is indicative of web browser-based lookup toolbars, which are the most commonly used anti-phishing method [77, 3]. The detection performance of the three tools on the 45 website pool used in the experiment was comparable to the results presented in Table 4. Details regarding the tools' detection performances are presented in Appendix F.

The anti-phishing tool was triggered each time the user clicked on any of the 10 URLs presented. The tool evaluated the website and made a recommendation. If the tool considered the website to be a phish, the user's web browser was redirected to a warning page. The standard Microsoft Internet Explorer warning page was used because it is similar to ones used by other popular browsers such as Mozilla Firefox and Google Chrome. It is important to note that this same warning was used for all three tools to ensure that the only observable difference between tools was their performance (i.e., predictions and run times). When presented with a warning, participants had the option of either heeding the warning and returning to the URL list without visiting the site, or ignoring the warning and continuing on to the website (by clicking on a URL on the warning page). If the tool considered the website legitimate, the URL's page was displayed in the web browser.

Users were scored based on their performance regarding the decisions they made. More specifically, performance was evaluated based on users' decisions to differentiate legitimate websites from phish, decisions to visit or avoid websites, and willingness to transact with phishing sites [24, 15, 74]. The experiment users were 120 students from a large university in the United States. Each user was randomly assigned to one of the six experiment settings (i.e., one of the three tools and spoof or concocted phishing websites). Overall, each of the six settings had the same number of participants (20). Prior to the experiment, users were given instructions regarding the aforementioned experiment task.

Table 7 shows the experiment results for the six settings. The values depicted for overall accuracy and legitimate/phishing *f*-measure, precision, and recall are averages based on users' decisions regarding the legitimacy of the 10 websites presented to them. The last two columns depict percentage of phishing websites actually visited by users (computed using web analytics software), and percentage of phishing websites that users were willing to transact with (i.e., opening a savings account). Based on the results, users who utilized the genre tree method as an anti-phishing tool significantly outperformed those who used AZProtect or IEFilter. The genre tree users' phishing detection recall was 10 percent and 6 percent higher than those using AZProtect on concocted and spoof websites, respectively. Consequently, considerably fewer users of the genre tree kernel visited phishing websites; 37–41 percent, versus over 60 percent for comparison methods. Moreover, 9 percent and 11 percent of genre tree users' total encounters with concocted and phishing websites resulted in a willingness to transact with a phishing website, respectively. Overall, users utilizing the genre tree kernel significantly outperformed those using AZProtect and IEFilter on legit recall, phish recall, phishing websites visited, and willingness to transact with phishing websites (on both the concocted and spoof settings). The one exception was legit recall on the concocted setting, where genre

Table 7. User Experiment Results for Genre Tree Kernel and Comparison Tools

		Legitimate websites			Phish detection			Visitation and willingness to transact	
Tool	Overall accuracy	F1	Prec.	Rec.	F1	Prec.	Rec.	Phish visited	Willingness to transact
Legitimate and concocted website settings									
Genre tree*	83.50	83.93	87.08	81.00+	84.24	82.56	86.00	37.00	9.00
AZProtect	77.50	78.80	78.61	79.00	78.30	80.75	76.00	61.00	14.00
IEFilter	63.00	62.59	62.19	63.00	64.44	65.94	63.00	82.00	25.00
Legitimate and spoof website settings									
Genre tree*	86.00	86.75	89.68	84.00	87.01	86.05	88.00	41.00	11.00
AZProtect	79.50	79.58	82.35	77.00	81.38	80.77	82.00	64.00	15.00
IEFilter	73.50	73.91	77.06	71.00	75.23	74.48	76.00	66.00	18.00

* Significantly outperformed both comparison tools, unless otherwise noted (*t*-test *p*-values appear in Appendix C, Table C7).

+ Did not significantly outperform AZProtect on legit recall for the concocted setting (*p*-value = 0.052).

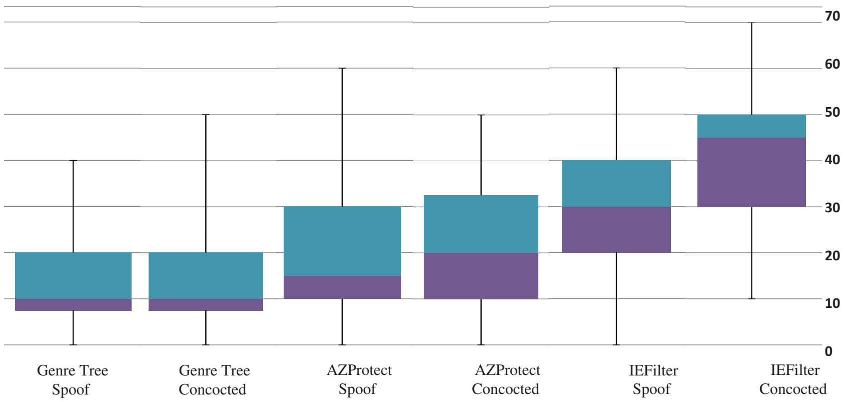


Figure 10. Percentage Disagreement (y-axis) between Users and Anti-phishing Tools

tree's performance was not significantly better (see Appendix C, Table C7 for t-test p-values).

Prior studies have noted that users often disregard tool warnings [74, 8]. Figure 10 presents box plots that depict the percentage disagreement between the three anti-phishing tools and their respective users, on the concocted and spoof experiment settings. For the genre tree kernel, the median disagreement was 10 percent and the third quartile was 20 percent on both experiment settings. In other words, at least 50 percent of users heeded the tool's warnings 90 percent or more of the time and at least 75 percent followed the tool's recommendations at least 80 percent of the time. Conversely, AZProtect's median disagreement rates were 15 percent and 20 percent on the spoof and concocted settings, respectively, and with third quartile values at or above 30 percent. In summary, users were 50–100 percent more likely to disregard AZProtect's warnings than those of the genre tree method, suggesting that the relationship between tool performance and users' likeliness to adhere to tool recommendations might be nonlinear. In the case of IEFilter, the disagreement levels were even more pronounced with medians of 30 percent on spoof and 45 percent on concocted websites. This result is consistent with prior studies, where users employing browser toolbars have been found to frequently disregard tool warnings [74, 15].

The results of the user study suggest that the enhanced performance of genre tree did indeed cause users to heed its warnings more, relative to AZProtect and IEFilter. Hence, the improved performance of the genre tree kernel was able to reduce the "cry-wolf" effect. This resulted in significantly better user phishing detection rates, less visitation of phishing websites, and lower willingness to transact with phishing websites. However, the results also underscore the need for additional work geared toward further strengthening the "user link" in the security chain. In particular, it is worth noting that user recall rates for legitimate and phishing sites were 10–15 percent lower than the performance of the tools utilized.

Discussion of Results

Our research objective for this study was to develop an IT artifact (method) that detected phishing websites with demonstratively better performance than that obtained by existing methods. To achieve this objective, we incorporated principles from genre theory. Our evaluation of the artifact highlights its marked improvement over existing state-of-the-art methods with respect to phishing detection rates, generalizability, run times, and effectiveness in user settings.

Experiments 1 and 2 demonstrated the enhanced accuracy of the genre tree kernel over existing content-based classifiers and toolbars, across various categories of spoof and concocted websites. Furthermore, by employing genre information, the proposed method was able to detect phishing websites in a computationally faster manner, and with better detection rates across various industry sectors (i.e., better generalizability), than the comparison content-based methods, which require the extraction of thousands of category-specific text, image, and linkage attributes. Experiment 3 revealed that methods utilizing genre information were often more effective than content-based methods, and that the genre tree kernel was also more effective than alternate genre- and/or tree-based methods. These findings further reinforce the efficacy of the proposed genre theoretic anti-phishing method. Experiment 4 showed that in comparison to benchmark methods, users utilizing the proposed method were far more likely to heed the tool's warnings (in some cases two to four times as likely as comparison tools). Consequently, users of the proposed method were able to better differentiate legitimate websites from phish, better avoid visiting phishing websites, and were less likely to transact with phishing websites. Given the substantial monetary and social costs that phishing websites continue to exert, the results have important implications insofar as improved anti-phishing methods remain an area of paramount importance.

Conclusions

In this study, we developed an innovative IT artifact that leverages genre theoretic principles for enhanced detection of phishing websites. The artifact was rigorously and extensively evaluated on a large testbed. Consistent with design science principles [27], we used a series of experiments to rigorously test the genre tree kernel method against existing state-of-the-art methods, against alternate genre- and tree-based methods, and in user settings. The experimental results revealed that the genre tree kernel was significantly more accurate, with legitimate, concocted, and spoof recall rates above 97 percent, 94 percent, and 99 percent, respectively. The results also showed that the genre tree kernel's performance was fairly consistent across website categories; of the 25 legitimate, concocted, and spoof categories examined in Figure 8, the genre tree kernel had the highest recall on 20 categories, and was significantly better than the best comparison method on 16 (ranging from financial, escrow, legal, and retail, to pharmacy, hospital, and health). Furthermore, the genre tree kernel was also significantly faster than the most accurate comparison method.

The enhanced performance also translated into better user security behavior; users employing the genre tree kernel were significantly better at avoiding phishing attacks. Overall, the results confirm the viability of using genre information for enhanced phishing detection.

Our research contributions are manifold:

- The development of a novel method for detecting phishing websites that fuses genre information and website design structure using a kernel-based classification technique. The artifact construction and evaluation process were guided by design science.
- The extensive evaluation of the proposed genre tree kernel and existing anti-phishing methods on a large-scale testbed encompassing over 4,000 legitimate and phishing websites. The evaluation included analysis of performance across various legitimate, concocted, and spoof industry sectors including financial, legal, retail, shipping, escrow, social networking, university, search engine, pharmacy, hospital, health-related sectors, and so on. The results revealed that performance for existing content-based methods varies considerably across industry sectors, suggesting that a "one size fits all" approach is impractical when relying heavily on text- and image-based attributes. To the best of our knowledge, this is the largest anti-phishing benchmarking study to date with respect to the number of tools, types of phishing attacks, and range of industry sectors examined.
- The results from a user study, which further highlighted the utility of the proposed method. While prior research has emphasized users' lack of trust and usage of anti-phishing tools [74, 15], the results of this work lend credence to the notion that more accurate security decision-support tools reduce user disregard rates, causing users to make better, more informed decisions. The findings suggest that future work on improving phishing detection methods is warranted, with potential to further thwart the "return on phishing" bottom line.
- The attainment of further insights from the user study. Given that users still underperformed the tools they were provided, the findings suggest that despite the improvements garnered by using the proposed genre-based method, user-tool dissonance remains. Getting users to heed tool warnings may require a multipronged approach, which encompasses enhancing tool detection performance, developing better warning delivery mechanisms [26], and providing effective education and training [41].

Given the importance of computer-aided credibility assessment [31, 32, 33, 69], the results of our work have important implications for various stakeholder groups, including households and organizations. In household settings, the median monetary cost associated with phishing-based identity theft is over $3,000 per victim [50]. In enterprise settings, a successful attack can cost over $1 million on average in recovery-related expenses alone; this number excludes hefty reputation costs, which are often difficult to quantify [28]. As phishing remains an omnipresent, proverbial thorn impacting organizations and society as a whole, better anti-phishing strategies remain a necessary endeavor. By taking a design science perspective to develop an IT artifact capable of markedly improving phishing detection rates and users' security-related decisions when encountering phishing attacks, this study constitutes an important step toward a more holistic, robust anti-phishing strategy.

Acknowledgements: This work was funded by the following grants from the U.S. National Science Foundation: CNS-1049497 and ACI-1443019. Research reported in this study is also partially supported by NSFC #71025001, #91024030, #71272236; and BJNSF #4132072. We would also like to thank our collaborators at McAfee Security for their guidance in developing the controlled experiment for the anti-phishing-tool user study.

References

1. Abbasi, A., and Chen, H. A comparison of fraud cues and classification methods for fake escrow website detection. *Information Technology and Management*, 10, 2–3 (2009), 83–101.

2. Abbasi, A., and Chen, H. A comparison of tools for detecting fake websites. *IEEE Computer*, 42, 10 (2009), 78–86.

3. Abbasi, A.; Zhang, Z.; Zimbra, D.; Chen, H.; and Nunamaker, J.F. Jr. Detecting fake websites: The contribution of statistical learning theory. *MIS Quarterly*, 34, 3 (2010), 435–461.

4. Abbasi, A.; Zahedi, F.M.; and Kaza, S. Detecting fake medical web sites using recursive trust labeling. *ACM Transactions on Information Systems*, 30, 4 (2012), no. 22.

5. Airoldi, E., and Malin, B. Data mining challenges for electronic safety: The case of fraudulent intent detection in e-mails. In *Proceedings of Workshop on Privacy and Security Aspects of Data Mining*. Brighton, 2004, pp. 57–66.

6. An, B. 14 arrested for making, selling fake drugs via bogus military medical websites. February 2, 2010. Available at http://english.people.com.cn/90001/90776/90882/6889316.html.

7. Catan, T. Google forks over settlement on rx ads. August 25, 2011. Available at http://online.wsj.com/news/articles/SB10001424053111904787404576528332418595052.

8. Zahedi, F.M.; Abbasi, A.; and Chen, Y. Fake-Website Detection Tools: Identifying Design Elements that Promote Individuals' Use and Enhance their Performance. *Journal of the Association for Information Systems*, forthcoming. http://aisel.aisnet.org/jais/forthcoming.html

9. Chou, N.; Ledesma, R.; Teraguchi, Y.; Boneh, D.; and Mitchell, J.C. Client-side defense against web-based identity theft. In *Proceedings of the Network and Distributed System Security Symposium*. San Diego, 2004, pp. 1–16.

10. Chua, C.E.H., and Wareham, J. Fighting Internet auction fraud: An assessment and proposal. *IEEE Computer*, 37, 10 (2004), 31–37.

11. Collins, M., and Duffy, N. Convolution kernels for natural language. In T.G. Dietterich, S. Becker, and Z. Ghahramani (eds.), *Advances in Neural Information Processing Systems* 14. Cambridge, MA: MIT Press, 2001, pp. 625–632.

12. Crimaldi, L. Google settles online pharmacy ad probe. August 24, 2011. Available at www.nbcnews.com/id/44257179/ns/business-us_business/t/google-settles-online-pharmacy-ad-probe/.

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

14. Cyr, D. Modeling website design across cultures: Relationships to trust, satisfaction and e-loyalty. *Journal of Management Information Systems*, 24, 4 (2008), 47–72.

15. Dhamija, R.; Tygar, J.D.; and Hearst, M. Why phishing works. *In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. Montreal, 2006, pp. 581–590.

16. Dinev, T. Why spoofing is serious Internet fraud. *Communications of the ACM*, 49, 10 (2006), 76–82.

17. Drost, I., and Scheffer, T. Thwarting the nigritude ultramarine: Learning to identify link spam. In *Proceedings of the European Conference on Machine Learning*. Porto, 2005, pp. 96–107.

18. Edworthy, J. Cognitive compatibility and warning design. *International Journal of Cognitive Ergonomics*, 1, 3 (1997), 193–209.

19. Ester, M.; Kriegel H.; and Schubert, M. Web site mining: a new way to spot competitors, customers, and suppliers in the World Wide Web. In *Proceedings of the ACM SIGKDD*. Edmonton, 2002, pp. 249–258.

20. Fu, A.Y.; Liu, W.; and Deng, X. Detecting phishing web pages with visual similarity assessment based on earth mover's distance (EMD). *IEEE Transactions on Dependable and Secure Computing*, 3, 4 (2006), 301–311.

21. Gartner, *Magic quadrant for web fraud detection*, May 30, 2013. Available at www.gartner.com/doc/2501221/magic-quadrant-web-fraud-detection.

22. Gaudinat, A.; Grabar, N.; and Boyer, C. Machine learning approach for automatic quality criteria detection of health web pages. In K.A. Kuhn, J.R. Warren, and T.-Y. Leong (eds.), *Proceedings of the World Congress on Health (Medical) Informatics: Building Sustainable Health Systems 129*. Amsterdam: IOS Press, 2007, pp. 705–729.

23. Goel, S.; Hofman, J.M.; and Sirer, M.I. Who does what on the web: A large-scale study of browsing behavior. In *Proceedings of the Sixth International Conference on Weblogs and Social Media*. Dublin, 2012, pp. 130–137.

24. Grazioli, S., and Jarvenpaa, S.L. 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 (2000), 395–410.

25. Gyongyi, Z., and Garcia-Molina, H. Spam: It's not for inboxes anymore. *IEEE Computer*, 38, 10 (2005), 28–34.

26. Herzberg, A., and Jbara, A. Security and identification indicators for browsers against spoofing and phishing attacks. *ACM Transactions on Internet Technology*, 8, 4 (2008), 1–36.

27. Hevner, A.R.; March, S.T.; Park, J., and Ram, S. Design science in information systems research. *MIS Quarterly*, 28, 1 (2004), 75–105.

28. Hong, J. The state of phishing attacks. *Communications of the ACM*, 55, 1 (2012), 74–81.

29. Huang, H.; Zhong, S.; and Tan, J. Browser-side countermeasures for deceptive phishing attack. In *Proceedings of the Conference on Information Assurance and Security*. Xi'an, 2009, pp. 352–355.

30. Jagatic, T.N.; Johnson, N.A.; Jakobsson, M.; and Menczer, F. Social phishing. *Communications of the ACM*, 50, 10 (2007), 94–100.

31. Jensen, M.L.; Lowry, P.B.; Burgoon, J.K.; and Nunamaker, J.F., Jr. Technology dominance in complex decision making: The case of aided credibility assessment. *Journal of Management Information Systems*, 27, 1 (2010), 175–202.

32. Jensen, M.L.; Lowry, P.B.; and Jenkins, J.L. Effects of automated and participative decision support in computer-aided credibility assessment. *Journal of Management Information Systems*, 28, 1 (2011), 201–233.

33. Jensen, M.L.; Averbeck, J.M.; Zhang, Z.; and Wright, K.B. Credibility of anonymous online product reviews: A language expectancy perspective. *Journal of Management Information Systems*, 30, 1 (2013), 293–323.

34. Joachims, T. Making large-scale SVM learning practical. In B. Scholkopf, C. Burges, and A. Smola (eds.), *Advances in Kernel Methods-Support Vector Learning*. Cambridge, MA: MIT Press, 1999, pp. 169–184.

35. Johnston, A.C., and Warkentin, M. Fear appeals and information security behaviors: An empirical study. *MIS Quarterly*, 34, 3 (2010), 549–566.

36. Juola, P., and Baayen, H. A Controlled-corpus experiment in authorship identification by cross-entropy. *Literary and Linguistic Computing*, 20, Supp. 1 (2005), 59–67.

37. Kaushik, A. *Web Analytics 2.0: The Art of Online Accountability and Science of Customer Centricity.* Indianapolis: Wiley, 2011.

38. Kashima, H.; Tsuda, K.; and Inokuchi, A. Marginalized kernels between labeled graphs. In *Proceedings of the Twentieth International Conference on Machine Learning*. Washington, DC,2003, pp. 321–328.

39. Kaspersky Lab. The evolution of phishing attacks: 2011–2013, 2013. Available at http://media.kaspersky.com/pdf/Kaspersky_Lab_KSN_report_The_Evolution_of_Phishing_Attacks_2011-2013.pdf.

40. Kirda, E., and Kruegel, C. Protecting users against phishing attacks. *Computer Journal*, 49, 5 (2006), 554–561.

42. Kumaraguru, P.; Sheng, S.; Aquisti, A.; Cranor, L.F.; and Hong, J. TeachingJohnny not to fall for phish. *ACM Transactions on Internet Technology*, 10, 2 (2010), no. 7.

42. Kwon, O., and Lee, J. Text categorization based on k-nearest neighbor approach for web site classification. *Information Processing and Management*, 39, 1 (2003), 25–44.

43. Le, A.; Markopoulou, A.; and Faloutsos, M. PhishDef: URL names say it all. In *Proceedings of the IEEE International Conference on Computer Communications*. Shanghai, 2011, pp. 191–195.

44. Li, J.; Zhang, Z.; Li, X.; and Chen, H. Kernel-based learning for biomedical relation extraction. *Journal of the American Society for Information Science and Technology*, 59, 5 (2008), 756–769.

45. Li, X.; Chen, H.; Zhang, Z.; Li, J.; and Nunamaker, J.F., Jr. Managing knowledge in light of its evolution process: An empirical study on citation network-based patent classification. *Journal of Management Information Systems*, 26, 1 (2009), 129–153.

46. Liu, W.; Deng, X.; Huang, G.; and Fu, A. Y. An antiphishing strategy based on visual similarity assessment. *IEEE Internet Computing*, 10, 2 (2006), 58–65.

47. Lim, C.S.; Lee, K.J.; and Kim, G.C. Multiple sets of features for automatic genre classification of web documents. *Information Processing and Management*, 41, 5 (2005), 1263–1276.

48. Lodhi, H.; Saunders, C.; Shawe-Taylor, C.; Cristianini, N.; and Watkins, C. Text classification using string kernels. *Journal of Machine Learning Research*, 2 (2002), 419–444.

49. March, S.T., and Smith, G. Design and natural science research on information technology. *Decision Support Systems*, 15, 4 (1995), 251–266.

50. McAfee. McAfee threats report: Fourth quarter 2010. February 8, 2011. Available at www.mcafee.com/us/resources/reports/rp-quarterly-threat-q4-2012.pdf.

51. Muller, K.; Mika, S.; Ratsch, G.; Tsuda, K.; and Scholkopf, B. An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks*, 12, 2 (2001), 181–201.

52. Ntoulas, A.; Najork, M.; Manasse, M.; and Fetterly, D. Detecting spam web pages through content analysis. *In Proceedings of the International World Wide Web Conference*. Edinburgh, 2006, pp. 83–92.

53. Nunamaker, J.F., Jr. Build and learn, evaluate and learn. *Informatica*, 1, 1 (1992), 1–6.

54. Nunamaker, J. F., Jr.; Chen, M.; and Purdin, T.D.M. Systems development in information systems research. *Journal of Management Information Systems*, 7, 3 (1991), 89–106.

55. Orlikowski, W.J., and Yates, J. Genre repertoire: The structuring of communicative practices in organizations. *Administrative Sciences Quarterly*, 39, 4 (1994), 541–574.

56. Pan, Y., and Ding, X. Anomaly based web phishing page detection. In *Proceedings of the Twenty-Second Annual Computer Security Applications Conference*. Miami Beach, 2006, pp. 381–392.

57. Phishtank.com. Phishing stats. 2013. Available at www.phishtank.com/stats/2013/01/.

58. Porter, M.E. An algorithm for suffix stripping. In K. Jones and P. Willett (eds.), *Readings in Information Retrieval*. San Francisco: Morgan Kaufmann, 1997, pp. 313–316.

59. Prince, B. Phishing attacks cost millions despite low success rates. December 7, 2009. Available at www.eweek.com/c/a/Security/Phishing-Attacks-Cost-Millions-Despite-Low-Success-Rate-879602/.

60. Ramzan, Z., and Wuest, C. Phishing attacks: Analyzing trends in 2006. In *Proceedings of the Fourth Conference on Email and Anti-Spam*. Mountain View, 2007. http://ceas.cc/2007/

61. Rosso, M.A. User-based identification of web genres. *Journal of the American Society for Information Science and Technology*, 59, 7 (2008), 1053–1072.

62. Roussinov, D.; Crowston, K.; Nilan, M.; Kwasnik, B.; Cai, J.; and Liu, X. Genre based navigation on the Web. In *Proceedings of the Thirty-Fourth Hawaii International Conference on Systems Sciences*. Maui, 2001. DOI:10.1109/HICSS.2001.926478

63. Saunders, C.; Shawe-Taylor, J.; and Vinokourov, A. String kernels, fisher kernels, and finite state automata. In S. Becker, S. Thrun, and K. Obermayer (eds.), *Advances in Neural Information Processin*g Systems 15. Cambridge, MA: MIT Press, 2002, pp. 633–640.

64. Schneier, B. Inside risks: Semantic network attacks. *Communications of the ACM*, 43, 2 (2000), 168.

65. Selinger, M. Google vs. Bing: Search engines deliver infected websites as their top results. 2013. Available at www.av-test.org/en/news/news-single-view/google-vs-bing-search-engines-deliver-infected-websites-as-their-top-results/.

66. Shepherd, M.; Watters, C.; and Kennedy, A. CyberGenre: Automatic identification of home pages on the web. *Journal of Web Engineering*, 3, 3–4 (2004), 236–251.

67. Storey, V.; Burton-Jones, A.; Sugumaran, V.; and Purao, S. Conquer: A methodology for context-aware query processing on the World Wide Web. *Information Systems Research*, 19, 1 (2008), 3–25.

68. Takimoto, E., and Warmuth, M.K. Path kernels and multiplicative updates. *Journal of Machine Learning Research*, 4 (2003), 773–818.

69. Twyman, N.W.; Elkins, A.C.; Burgoon, J.K.; and Nunamaker, J.F., Jr. A rigidity detection system for automated credibility assessment. *Journal of Management Information Systems*, 31, 1 (2014), 173–201.

70. Urvoy, T.; Chaveau, E.; Filoche, P.; and Lavergne, T. Tracking web spam with hidden style similarities. *ACM Transactions on the Web*, 2, 1 (2008), no. 3.

71. Willis, P. Fake anti-virus software catches 43 million users' credit cards. October 20 2009. Available at www.digitaljournal.com/article/280746/.

72. Witten, I.H., and Frank, E. *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd ed. San Francisco: Morgan Kaufmann, 2005.

73. Wright, R.T., and Marett, K. The influence of experiential and dispositional factors in phishing: An empirical investigation of the deceived. *Journal of Management Information Systems*, 27, 1 (2010), 273–303.

74. Wu, M.; Miller, R.C.; and GarfunkelS., Do security toolbars actually prevent phishing attacks? In *Proceedings of the Conference on Human Factors in Computing Systems*. Quebec, 2006, pp. 601–610.

75. Xiang, G.; Hong, J.; Rose, C.P.; and Cranor, L. Cantina+: A feature-rich machine learning framework for detecting phishing web sites. *ACM Transactions on Information and System Security*, 14, 2 (2011), no. 21.

76. Yoshioka, T.; Herman, G.; Yates, J.; and Orlikowski, W. Genre taxonomy: A knowledge repository of communicative actions. *ACM Transactions on Information Systems*, 19, 4 (2001), 431–456.

77. Zhang, Y.; Egelman, S.; Cranor, L.; and Hong, J. Phinding phish: Evaluating anti-phishing tools. In *Proceedings of the Fourteenth Annual Network and Distributed System Security Symposium*. San Diego, 2007, February 28-March 2.

78. Zhang, Y.; Hong, J.; and Cranor, L. CANTINA: A content-based approach to detecting phishing web sites. In *Proceedings of the International World Wide Web Conference*. Banff, 2007, pp. 639–648.

Appendix A: Website Categories in Testbed

The testbed included websites associated with numerous industry sectors, including shipping, financial, escrow, legal, retail, search engine, social networking, university, pharmacy, health, and hospital websites. Table A1 shows the number of legitimate, concocted, and spoof websites pertaining to these sectors encompassed in the testbed. The composition of concocted and spoof websites in the test bed is somewhat comparable to the summary statistics presented in Figure 2 (in the main

Table A1. Number of Legitimate, Concocted, and Spoof Websites for Various Categories in the Testbed

Grouping and/or industry sector	Legitimate	Concocted	Spoof
EBay and PayPal*	2	0	200
Escrow*	60	350	200
Financial	200	400	400
Health	193	200	0
Hospital	130	25	0
Legal	100	100	0
Pharmacy	180	325	50
Retail+	250	50	225
Search engine/Social networking	25	0	200
Shipping+	10	100	50
University^	200	0	25
Total	1,350	1,350	1,350

* PayPal and escrow typically appear as part of the payment services industry sector in reports, and in Figure 2.

+ Retail and shipping are subgroups in the retail/services industry sector in reports, and in Figure 2.

^ University and search engine websites typically appear in the "other" category in reports, and in Figure 2.

document), which is based on reported incidents in online phishing databases such as Artists Against 419, PhishTank, and Anti-Phishing Working Group. For instance, concocted websites tend to concentrate heavily on the escrow (i.e., payment services), financial, health, legal, pharmacy, and shipping (i.e., part of retail/service) industry sectors. In contrast, spoof websites geared toward identity theft tend to focus on financial, escrow, retail, and search/social. The legitimate websites were those commonly spoofed as well as those associated with the concocted website categories.

It is important to note a few differences between the industry sectors reported in Figure 2 (based on fraud prevention community reports), and the groupings presented in Table A1. First, whereas eBay and PayPal belong to the auction sites and payment services industry sectors, respectively, these two websites have traditionally been targeted heavily. Consequently, many anti-phishing tools have developed fraud cues and detection rules either specifically based on, or at least partially inspired by, their spoof attacks. Examples include eBay Account Guard and SpoofGuard. Hence, such tools' performance on PayPal spoofs is often not indicative of their performance on other payment services concocted and spoof websites. When presenting the industry sector-level results in the evaluation section, we grouped these two together and placed escrow websites in its own group to better present performance differences for these two key payment services subgroups (i.e., PayPal and escrow websites). Second, we separated retail/services into its two major subgroups: retail and shipping. This was done since both subgroups are highly pervasive, and often exhibit significant differences with respect to content and fraud cues. Third, search engines were grouped with social networking websites, and university websites were made their own category (though both typically appear in the "other" category in quarterly reports from organizations such as the Anti-Phishing Working Group).

Appendix B: Evaluation Metrics

Class-level Recalls:

$$\text{Legit Recall} = \frac{\text{number of legitimate websites classified as legitimate}}{\text{total number of legitimate websites}}$$

$$\text{Concocted Recall} = \frac{\text{number of concocted websites classified as phish}}{\text{total number of concocted websites}}$$

$$\text{Spoof Recall} = \frac{\text{number of spoof websites classified as phish}}{\text{total number of spoof websites}}$$

Class-level Precisions:

$$\text{Legit Precision} = \frac{\text{number of legitimate websites classified as legitimate}}{\begin{array}{c}\text{number of legitimate classified as legitimate} + \\ \text{number of phish classified as legitimate}\end{array}}$$

$$\text{Concocted Precision} = \frac{\text{number of concocted websites classified as phish}}{\begin{array}{c}\text{number of concocted classified as phish} + \\ \text{number of legitimate classified as phish}\end{array}}$$

$$\text{Spoof Precision} = \frac{\text{number of spoof websites classified as phish}}{\begin{array}{c}\text{number of spoof classified as phish} + \\ \text{number of legitimate classified as phish}\end{array}}$$

Class-level F-measures:

$$\text{Legit F} - \text{measure} = \frac{2 \times \text{Legit Precision} \times \text{Legit Recall}}{\text{Legit Precision} + \text{Legit Recall}}$$

$$\text{Concocted F} - \text{measure} = \frac{2 \times \text{Concocted Precision} \times \text{Concocted Recall}}{\text{Concocted Precision} + \text{Concocted Recall}}$$

$$\text{Spoof F} - \text{measure} = \frac{2 \times \text{Spoof Precision} \times \text{Spoof Recall}}{\text{Spoof Precision} + \text{Spoof Recall}}$$

Overall Accuracy:

$$\text{Overall Accuracy} = \frac{\text{number of correctly classified legitimate, concocted, and spoof websites}}{\text{total number of websites}}$$

Appendix C: Paired T-tests for Research Questions

Experiment 1:

Table C1. *P*-values for Paired *T*-tests Comparing Genre Tree Kernel Classification Performance Against Kernel and Non-kernel Content-based Methods (RQ1a)

	Comparison techniques									
	Kernel-based					Non-kernel-based				
Website categories	Linear comp.	Linear	Poly-2	RBF	Poly-3	Logit Reg.	J48 DT	Bayes Net	Naive Bayes	Neural net
Legitimate (n = 1350)	<0.001*	<0.001*	<0.001*	<0.001*	<0.001*	<0.001*	<0.001*	<0.001*	<0.001*	<0.001*
Concocted (n = 1350)	<0.001*	<0.001*	<0.001*	<0.001*	<0.001*	<0.001*	<0.001*	<0.001*	<0.001*	<0.001*
Spoof (n = 1350)	<0.001*	<0.001*	<0.001*	<0.001*	<0.001*	<0.001*	<0.001*	<0.001*	<0.001*	<0.001*

* *P*-values significant at alpha = 0.05.

Table C2. *P*-values for Paired *T*-tests Comparing Genre Tree Kernel Classification Performance Against Best Comparison Content-based Method (Linear Composite Kernel) Across Industry Sectors (RQ1b)

	Industry sector or website group										
Website categories	eBay PayPal	Escrow	Finance	Health	Hosp.	Legal	Pharm.	Retail	Search social	Ship.	Univ.
Legitimate	—	0.022*	0.042*	0.027*	<0.001*	0.208	0.003*	0.017*	0.081	—	0.042*
Concocted	—	<0.001*	<0.001*	0.031*	0.332	0.235	<0.001*	0.161	—	0.014*	—
Spoof	0.007*	0.001*	<0.001*	—	—	—	0.079	<0.001*	0.159	0.161	0.500

* *P*-values significant at alpha = 0.05.

Experiment 2:

Table C3. Paired *T*-test *P*-values for Genre Tree Kernel Detection Versus Anti-phishing Tools (RQ2a)

	Comparison anti-phishing tools							
	Classifier systems							
Website categories	AZProtect	CANT INA+	Net Craft	Norton SW$^+$	PhishDef	CANTINA	Spoof Guard	eBay AG
Legitimate (*n* = 1,350)	<0.001*	<0.001*	<0.001*	<0.001*	<0.001*	<0.001*	<0.001*	<0.001*
Concocted (*n* = 1,350)	<0.001*	<0.001*	<0.001*	<0.001*	<0.001*	<0.001*	<0.001*	<0.001*
Spoof (*n* = 1,350)	<0.001*	<0.001*	<0.001*	<0.001*	<0.001*	<0.001*	<0.001*	<0.001*

	Lookup systems			
	IEFilter	FirePhish	Site Hound	EarthLink
Legitimate (*n* = 1,350)	<0.001*	<0.001*	<0.001*	<0.001*
Concocted (*n* = 1,350)	<0.001*	<0.001*	<0.001*	<0.001*
Spoof (*n* = 1,350)	<0.001*	<0.001*	<0.001*	<0.001*

+ Norton Safe Web is a hybrid tool that combines content-based signatures with a community-driven blacklist.

* *P*-values significant at alpha = 0.05.

Table C4. P-values for Paired T-tests Comparing Genre Tree Kernel Classification Performance Against Best Comparison Anti-phishing Tool (AZProtect) Across Industry Sectors (RQ2b)

	Industry sector or website group										
Website categories	eBay PayPal	Escrow	Finance	Health	Hosp.	Legal	Pharm.	Retail	Search social	Ship.	Univ.
Legitimate	—	0.091	0.042*	0.015*	0.029*	0.048*	<0.001*	0.017*	0.081	—	—
Concocted	—	<0.001*	0.001*	0.039*	0.133	0.399	<0.001*	0.009*	—	0.026*	—
Spoof	0.022*	0.002*	<0.001*	—	—	—	0.079	0.012*	0.159	0.161	0.164

* *P*-values significant at alpha = 0.05.

Table C5. *P*-values for Paired *T*-tests Comparing Genre Tree Kernel Run Times Against Best Comparison Anti-phishing Tool (AZProtect) Across Industry Sectors (RQ2c)

	Industry sector or website group										
Website categories	eBay PayPal	Escrow	Finance	Health	Hosp.	Legal	Pharm.	Retail	Search social	Ship.	Univ.
Concocted	—	<0.001*	0.001*	<0.001*	<0.001*	<0.001*	<0.001*	<0.001*	—	<0.001*	—
Spoof	<0.001*	<0.001*	<0.001*	—	—	—	<0.001*	0.042*	<0.001*	<0.001*	0.310

* *P*-values significant at alpha = 0.05.

Experiment 3:

Table C6. P-values for Paired T-tests Comparing Genre Tree Kernel against Alternative Genre and Tree Kernel-based Methods (RQ3)

	Alternative genre and tree kernel-based methods							
	GT-RW		GT		G		T and T-RW	
Website categories	Cross Entropy	Con Path	Max Subtree	Num Subtree	Poly 3	Poly 2	Cross Entropy	Max Subtree
Legitimate (*n* = 1,350)	<0.001*	<0.001*	<0.001*	<0.001*	<0.001*	<0.001*	<0.001*	<0.001*
Concocted (*n* = 1,350)	<0.001*	<0.001*	<0.001*	<0.001*	<0.001*	<0.001*	<0.001*	<0.001*
Spoof (*n* = 1,350)	<0.001*	<0.001*	<0.001*	<0.001*	<0.001*	<0.001*	<0.001*	<0.001*

Note: Only the top two performing alternate methods from each category were compared against genre tree kernel.
* *P*-values significant at alpha = 0.05.

Experiment 4:

Table C7. *P*-values for Pair-wise *T*-tests Comparing User Performance When Utilizing Genre Tree Kernel Versus Alternate Tools in User Study (RQ4a-c)

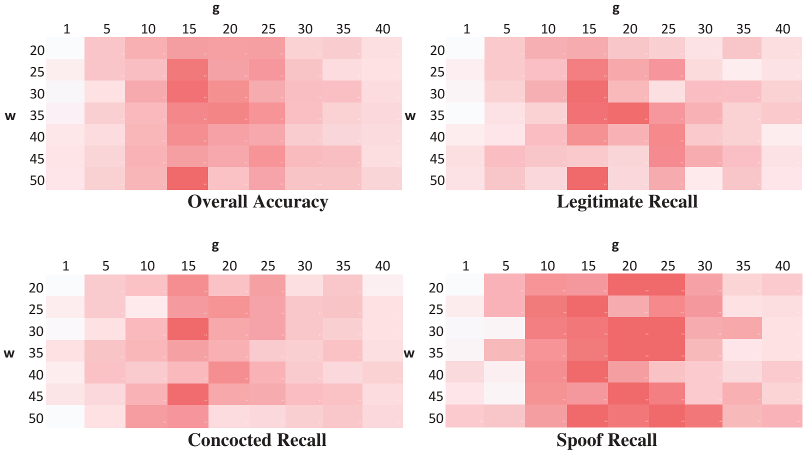
	Comparison anti-phishing tools							
	AZProtect				IEFilter			
Alternate tool	Legit recall	Phish recall	Phish visited	Willingness to transact	Legit recall	Phish recall	Phish visited	Willingness to transact
Concocted setting (*n* = 400)	0.052	<0.001*	<0.001*	0.047*	<0.001*	<0.001*	<0.001*	<0.001*
Spoof setting (*n* = 400)	<0.001*	<0.001*	<0.001*	<0.001*	<0.001*	<0.001*	<0.001*	<0.001*

* *P*-values significant at alpha = 0.05.

Appendix D: Impact of Parameter Settings

In order to assess the impact of different parameter settings on the genre tree kernel's performance, we ran various combinations of values for *w* (number of random walks performed) and *g* (maximum number of child nodes for a genre). For *w*, we ran values of 20 through 50 in increments of 5, while *g* was run using values of 1 and 5–40 in increments of 5 (resulting in 63 total combinations). Figure D1 shows heat maps for overall accuracy and legit, concocted, and spoof recall for all 63 parameter settings. Darker regions denote higher accuracy/recall rates. Based on the heat maps, it is evident that the genre tree kernel performed best for values of *g* ranging from 10 to 25, with the best performance attained when *g* was set to 15. In contrast, the *w* parameter did not seem to have any discernible pattern for values ranging from 20 to 50.

Table D1 shows the settings with the highest and lowest overall accuracies, as well as the setting utilized in the experiments (*g* = 10; *w* = 30). Interestingly, even the setting yielding the lowest results outperformed comparison methods. The results suggest that the genre tree kernel's performance is fairly robust across parameter settings.



Overall Accuracy

Legitimate Recall

Concocted Recall

Spoof Recall

Figure D1. Performance for Various Genre Tree Parameter Settings

Table D1. Performance Results for Best, Worst, and Used Parameter Settings for Genre Tree Kernel

		Legitimate websites (n = 1,350)			Concocted detection (n = 1,350)			Spoof detection (n = 1,350)		
Genre tree parameter setting	Overall accuracy (n = 4,050)	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.
Best (g = 15, w = 30)	98.15	97.26	95.96	98.59	97.30	98.56	96.07	99.19	98.61	99.78
Used (g = 10, w = 30)	96.91	95.44	94.03	96.89	95.50	96.80	94.22	98.28	96.97	99.63
Worst (g = 1, w = 20)	95.06	92.76	90.72	94.89	93.75	94.78	92.74	96.27	95.02	97.56

Appendix E: Genre and Level Composition of Legit, Concocted, and Spoof Websites

Figure E1 shows the number of genres and levels associated with each of the 4,050 websites in the training set (i.e., 1,350 legit, concocted, and spoof, respectively). Based on the figure, it is apparent that legitimate websites typically have more genres and levels as compared to concocted websites, and more genres and fewer levels as compared to spoof websites. Concocted tend to be shallow while spoofs are buried deeper on servers. Further breaking down by genres per level, as done by the G-linear method allows accuracies of around 90 percent. However, the proposed GT-RW-PM method's inclusion of structure in addition to genres and levels enables better performance (approximately 7 percent higher). As demonstrated in the user study (Experiment 4), the additional bump in accuracy enables GT-RW-PM to attain performance levels that are better aligned with users' acceptable fault tolerances for anti-phishing.

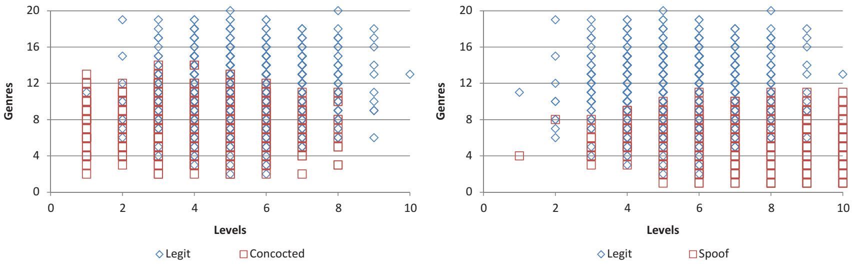


Figure E1. Genres and Site Levels in Legit, Concocted, and Spoof Websites

Appendix F: Performance of Anti-phishing Tools on Bank Websites Used in User Experiment

For the user study, a pool composed of 15 legitimate, 15 concocted, and 15 spoof commercial bank websites was incorporated into the experiment. Each user was randomly assigned 5 legitimate and 5 concocted or spoof websites. All three anti-phishing tools were run on the 45 websites. The three tools' detection performances are presented in Table E1 (i.e., percentage recall). For all three tools, the performance on these 45 websites was comparable to results attained across all financial websites in the testbed.

Table E1. Performance of Genre Treel, AZProtect, and IEFilter on 45 Websites used in User Experiment

Website category	Legitimate recall	Concocted recall	Spoof recall
Genre tree	100	93	100
AZProtect	93	86	93
IEFilter	100	20	73