Detection of Fraudulent Websites Through Third-party Request Structure

Ram D. Gopal[a], Afrouz Hojati[b,*], Raymond A. Patterson[b]

[a] *Warwick Business School, University of Warwick, Scarman Rd, Coventry, CV4 7AL, United Kingdom*
[b] *Haskayne School of Business, University of Calgary, 2500 University Dr NW, Calgary, T2N 1N4, AB, Canada*

Abstract

Third-party websites or applications are the key entities in the web eco-system that enable websites to function and offer services. Almost every organization today uses dozens of websites and sub-domains. Each provides essential functions and typically uses dozens of third-parties to produce its capabilities. With the growing problem of illegitimate websites, such as those peddling fake news and selling counterfeit products, the detection of fraudulent websites becomes more and more crucial. While the conventional method of fraudulent website detection mostly relies on the content-based analysis of websites, the method of this study uses third-party request structure features and attributes of third-parties engaged in the structure to predict legitimate and fraudulent websites. This method can be used on a real-time basis to complement current detection methods. Moreover, our approach is not limited to a specific category of websites. In other words, unlike previous studies, our approach is able to increase the likelihood of detecting all kinds of fake and fraudulent websites. The results of this study are largely robust across different predictive models.

Keywords: Fraudulent Website Detection, Third-party, Prediction, Machine Learning

*Corresponding author
Email address: Afrouz.Hojati@ucalgary.ca (Afrouz Hojati)

1. Introduction

The evolution of the internet has brought with it a number of extremely convenient advances in our daily life. It has also given way to new risks and increased concerns about fake, fraudulent, and illegitimate activities over the Internet. The cybercrime industry's revenue has grown to $1.5 trillion annually, with $860 billion of that (approximately 57%) being related to illegal online markets [1]. Moreover, Amazon alone spent $500 million in 2019 to fight fraud, abuse, and counterfeit products [2]. Fake, fraudulent, and illegitimate activities over the Internet are not limited to counterfeit products. According to the FBI's Internet Crime Complaint Center (IC^3), $26 billion in losses were related to scam and phishing activities globally between June 2016 and July 2019 [3]. These numbers do not include the societal impact of other internet-based fraudulent and misleading activity such as fake news and clickbait. Identifying nefarious websites that deal in fraudulent activities is a first and critically important step in combating them. In this study, we aim to provide mechanisms to detect websites with fake, fraudulent, and illegitimate activities over the Internet, and we collectively refer to these websites as fraudulent websites in this paper.

Detecting fraudulent websites is typically based on website content, blacklists, or consumer complaints. In contrast to previous studies that mostly rely on content-based methods (including textual content, design, components, and metadata of a website) for detection, our approach uses the infrastructure structure behind the scene for detection of legitimate versus fraudulent websites.

For a website or a web-based application to operate, it has a variety of requirements or services that are offered by other vendors called third-parties, rather than the website itself. Gopal et al. [4] refer to third-party supply chains "as digital supply chains, where content and services are supplied from upstream third-parties to downstream websites." Our approach utilize the structure that third-parties working with or serving a given website would create. This structure is based on the requests sent by the website to third-parties or sent by

third-parties to other third-parties. Legitimate websites extensively use third-parties to obtain services related to functionality, performance improvement, and targeting and advertising. Fraudulent websites are no exception. Some widespread third-party services include developer utilities, social networks widgets, search engines, and advertising and analytic features. Figure 1 shows how a website (blue circle in the middle) is connected to different third-parties. These connections, which are the requests for various services, can be made either directly by the website to a third-party, or indirectly from one third-party to another. All these requests together form a "request structure" which shows the flow of all requests, including the sender and receiver of each request (see Figure 1 and 2). The request structure is also known as the "third-party supply chain", "third-party network" or "third-party request network". Further details on the request structure are presented in section 3.

The third-party request network is important for security, functionality, and performance of websites, and it also impacts the risk of leaking personal information (often intentionally). This third-party request structure is the artifact of the website's business model. In this study, we show that the third-party request structure can be effectively used to differentiate between fraudulent versus legitimate websites. Using features of the third-party request structure, we are able to increase the likelihood of detecting fraudulent websites across a variety of categories (e.g. news, e-commerce, and phishing websites). The ability to create generalized detection algorithms to detect fraudulent websites, regardless of website category, is a key contribution of this paper. A variety of machine learning predictive models are used to provide robustness of results.

The remainder of this paper is structured as follows. Section 2 presents background on previously proposed approaches and methods of the detection of fake or fraudulent websites. Section 3 will provide more information on how websites work and their relation with third-parties. For a better understanding of the third-party request structure, a detailed example is provided in this section. Our methodology is described in section 4 and section 5 presents and discusses the computational results. Finally, section 6 presents the conclusion,

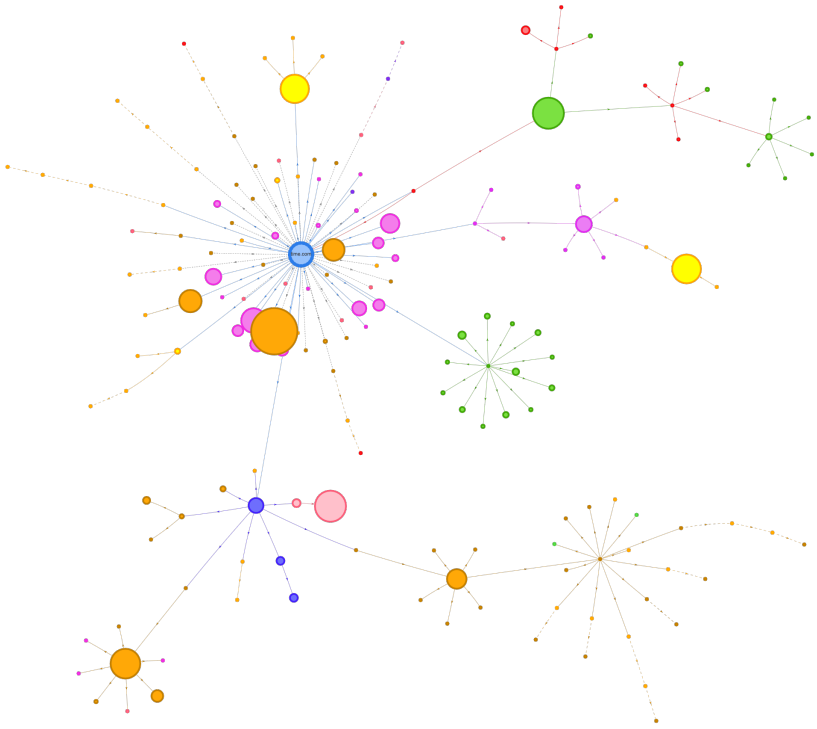

Figure 1: Request structure of a website (Time.com)

limitations, and suggestions for future work.

2. Literature Review

In this section we first discuss the two types of fraudulent website detection methods, lookup and classification mechanisms. Then, we provide a review of research on four prominent categories of fraudulent websites including phishing, fake news, fraudulent e-commerce, and piracy websites. Then, we will discuss the shortcoming of existing methods and how the proposed method of this study address those shortcoming.

2.1. Lookup Mechanisms

For identifying fake or fraudulent websites, two mechanisms are used: lookup and classification mechanisms. In the lookup mechanism, many URLs of fake

or fraudulent websites are collected and stored in a blacklist based on multiple confirmed sources. Legitimate URLs can also be stored in a white list. Many lookup systems also allow users to report websites directly through their interface. Then, by checking whether the requested website exists in the blacklists or not, lookup systems determine the legitimacy of the website [5, 6]. Blacklists are created based on various categories of websites. The Anti-Phishing Working Group (APWG) and Phishtank.com are two sources for phishing websites blacklists. Wikipedia and Harvard University Library created lists of fake news websites[7, 8]. Although, previous studies have empirically demonstrated that lookup systems have very high performance in detection of fraudulent websites, they are naturally incomplete. Because many new fraudulent websites are continuously created and it takes at least a couple of hours on average before they are verified in the blacklists [9]. In response, classification systems are widely used to deal with this problem.

2.2. Classification Mechanisms

Classification mechanisms mainly rely on appearance of fraud cues in the website to discern between legitimate and fraudulent websites. Research mainly use features related to URL and links contained within a page (e.g. length of full URL, domain name, directory, file and the number of symbols) to evaluate the likelihood whether the website is fraudulent or not[10, 11, 12, 13]. Text-based features including spellings and grammatical mistakes, lexical measures (e.g., total words per page, average sentence length), and the frequency of certain word phrases were used along with the URL-based features to detect fraudulent websites[14, 15]. Over the time, many studies add new features to the existing features in order to enhance performance of detection methods including image-based, linkage-based, source code- and HTML-based, style-based, search-based, and domain registration-based features. Abbasi et al. [9] categorized all these features as "Content-Based" features. We provide a brief literature review on content-based classification methods for four prominent categories of fraudulent websites including phishing, fake news, fraudulent e-commerce, and piracy

websites. A summary of previous studies are available in Table 1.

- **Phishing Websites**

  Phishing websites mainly refer to websites engage in identity theft by mimicking legitimate websites. The proposed method by Teraguchi and Mitchell [10], SpoofGuard, uses basic criteria mostly based on URL to check whether a website is phishing or not. Wu et al. [16] extract features based on the domain registration information (e.g. domain name, host name, host country, and date of registration) for prediction. CANTINA, proposed by Zhang et al. [14], uses text-based features for the detection of phishing websites. In this method, by using term frequency-inverse document frequency (TF-IDF), a web page's keywords are extracted and searched on Google. The website is classified as legitimate with this method if search results include the website name. In the enhanced version of CANTINA, CANTINA+ [17], many features other than text-based features such as features related to the URL and HTML of websites are used to improve the results. Abbasi et al. [18] use cues such as textual, URL, source code, images and linkage features, along with statistical learning theory (STL), to detect fake websites. Using STL, they are able to detect fake websites with 96% accuracy. Image-based features alone or along with textual features are mainly used in [15, 19, 20, 21] to assess visual similarities between phishing websites and legitimate websites. Wenyin et al. [22] use URL, links within the website, and body text of a website for detection of phishing websites and their targets based on construction and reasoning of their Semantic Link Network (SLN). Tan et al. [11] relies on URL features and search engine results for prediction. Using various classification algorithms, Yuan et al. [23] show that features related to URL and links within the webpage are helpful for phishing website detection. There exist huge numbers of research on phishing detection that use content-based features with different machine learning approach [24, 12, 25, 26, 13, 27]. Actually, the recent studies do not introduce

any new features, but they only make use of new machine learning approaches to improve prediction. For example, Yerima and Alzaylaee [27] propose a deep learning model based on Convolutional Neural Networks (CNN) for the detection while the features they use are the traditional URL and web content features. The only research we found that is trying to use new features for detection is conducted by Abbasi et al. [9]. They use fraud cues that are associated with differences in purpose between legitimate and phishing websites, manifested through genre composition and design structure of web pages for detecting phishing websites. Although, their features are not the same as prior studies, they extracted them from content of websites. Therefore, their features are also considered as content-based features which we will discuss drawbacks of using content-based features for predicting fraudulent websites in next section.

- **Fake E-commerce Websites**

  This category of fraudulent websites includes fraudulent online stores not delivering ordered products or selling counterfeit goods. Wadleigh et al. [28] analyse and predict websites selling counterfeit products by using URL, HTML, text, registration information, and Alexa ranking of websites. Maktabar et al. [29] proposes a fraudulent website detection model based on sentiment analysis of the textual contents of websites, using natural language processing and supervised machine learning techniques. Carpineto and Romano [30] propose a system for identifying fake e-commerce websites which uses content on a given websites, HTML features, and search engine result. Mostard et al. [31] use two types of features including HTML and linkage features of the website, along with the visual features extracting from logos and images to predict fraudulent online stores. In another instance of content-based detection, Kassim et al. [32] use three types of features (HTML tags, textual content, and the image of the website) for detection of fraudulent e-commerce websites. Utilizing machine learning algorithms such as Linear Regression, Decision Tree,

and Random Forest, they show that employing combined features result in improving the overall accuracy of the classifiers. In another attempt, using machine learning processes, Beltzung et al. [33] classify fraudulent online shops based on only source-code features (e.g. CSS, JavaScript and comments).

- **Fake News Websites**

  Fake news detection methods mainly use two types of features for detection including content-based features and social context-based features. Content based features mainly refers to textual or linguistic features which extract information from the news text and consist of syntactic, lexical (e.g. number of words and syllables per sentence, tags of word categories (such as noun, verb, adjective)), and semantic features. Social context-based features consist of statistics of user behaviour and network patterns from social media. These features are widely used in the previous studies for detection of fake news. For example, Castillo et al. [34] extracted features from text (e.g. hashtags, special characters, symbols, and sentiment), posting behaviour of users, and citations to external sources to classify news. Gupta et al. [35] use a set of forty-five features to access the credibility of user generated content on Twitter. The set contains features based on tweet meta-data, tweet content, users (number of followers, friends, etc.) and the network (e.g. number of retweets, mentions, reply, and etc.). Chen et al. [36] use content features such as semantics, as well as non-text features, such as image analysis and user behavior, to recognize false news content. Ahmed et al. [37] use text analysis based on n-gram features and machine learning classification to detect fake news. [38] propose semi-supervised learning model to identify a fake news on social media. They extracted the opinion expressed in the replies to the tweets about a given news, evaluated the credibility of the users who posted a tweet or replied about the news and evaluated the relationship between these users to determine either a given news is fake or real. Existing

methods do not distinguish real news websites from fake news websites, but rather, evaluate trustworthiness at the article level. In contrast, our proposed method focuses on detection of fake news websites rather than fake articles. The only paper that works on website detection of fake news is conducted by Gopal et al. [4]. They utilized third-parties relation as input to machine learning algorithms to discern between trustworthy and untrustworthy news websites with up to 95% accuracy. Similar to content-based methods, the results apply narrowly to a specific category of websites, which, in this case, was news websites.

- **Piracy Website**

  A great deal of research relates to detection of pirated digital content and product on the internet such as movies, audios, software, and games [39, 40, 41, 42, 43, 44, 45, 46, 47, 48]. However, fewer research exists on the detection of piracy websites that illegally copy and distribute such contents without the permission of the copyright owner. Studies such as [49] and [50] mainly use the body text of websites, linkages within a website, HTML source code, and advertisements on the websites to detect whether the website is a pirate or not.

2.3. Shortcomings of Content-based Classification Methods

Prior studies using content-based features yielded good results. However, they suffer from various shortcomings. The first shortcoming is *generalizability*. Features used in content-based methods are typically extracted from legitimate and fraudulent websites associated specifically with either a certain industry sector (e.g., news, financial, medical, etc.) or a specific category of websites (e.g., phishing or fake news). Specific content-based techniques applied to one category might prove to be less effective on websites from other categories. Detection methods that generalize across website categories would be strongly preferred. The second shortcoming is *computationally inefficiency*. Extracting thousands of attributes from hundreds of web pages per website can result in

Table 1: Literature Review

Fraud Cues (Features)	Research
Phishing Websites	
Body Text	Zhang et al. [14]
Body Text , HTML	Yang et al. [26]
Image, Body Text	Liu et al. [15], Zhang et al. [20]
Image	Fu et al. [19], Chiew et al. [21]
URL	Tan et al. [11], Bahnsen et al. [12], Ubing et al. [13]
URL, HTML, Domain Registration Information	Xiang et al. [17]
URL, Domain Registration Information	Le et al. [51]
URL, HTML, Body Text, URLS, Image, linkage	Abbasi et al. [18]
URL, HTML	Marchal et al. [24]
URL, Linkage	Teraguchi and Mitchell [10]
URL, Linkage, Body Text	Wenyin et al. [22], Yuan et al. [23]
Website Genres	Abbasi et al. [9]
Fake e-commerce Websites	
HTML, Image, Website Metadata	Mostard et al. [31]
HTML, Image, Body Text	Kassim et al. [32]
HTML, Body Text	Carpineto and Romano [30]
Body Text(Pricing Data)	Wadleigh et al. [28]
Source Code Similarity	Beltzung et al. [33]
Body Text	Maktabar et al. [29]
Fake News Websites	
Text, Social Context (Article detection)	Castillo et al. [34], Gupta et al. [35], Ahmed et al. [37]
	Ahmed et al. [37], Konkobo et al. [38], Reis et al. [52]
Text, Image, Social Context (Article detection)	Chen et al. [36], Shu et al. [53]
Third-party Partnership	Gopal et al. [4]
	(This is the only study to use non-content-based features)
Piracy Websites	
HTML, Linkage, Text, Ads	Choi and Kwak [49], Kim and Kwak [50]

computational inefficiencies. Lengthy run times related to content-based detection methods is unsuitable for real-time environments. Thus, computationally efficient detection methods are highly desirable. The third shortcoming is *adaptability*, or rather the lack thereof. Due to the evolving nature of websites, the features of legitimate websites evolve over time and fraudulent websites learn to evade detection. This evolution results in a continuous cat and mouse game of deception and detection [18, 9].

Previous literature is entirely content-based except for [4], and is typically applied only to a specific type of website. In this paper, we propose a new set

of website third-party features (i.e., not content-based) to facilitate generalized, computationally efficient, and adaptable machine learning algorithms to detect fraudulent websites. This work creates a new line of research attempting to find universally applicable algorithms that can serve as a first line of defence against the myriad of new fraudulent websites that pop up continuously. This work is not intended to supplant previous research related to specific contexts, but is instead intended to add another layer of defense for individuals who are surfing the web.

3. Third-party Request Structure

In order to access a website, a user's browser sends an HTTP request to a server using the URL to retrieve the web page the server should display. Then, numerous calls to various third-parties are embedded into the HTML code of the website. These HTTP calls ask for the services the website needs to do many of the things that it does. Each request can be either sent directly from the websites to a third-party or indirectly from third-parties to other third-parties. Much third-party activity occurs indirectly, and this is not included in the HTML code as only the first layer of direct calls are included in the HTML code. The structure of all requests sent to third-parties can be seen in Figure 1.

For a better understanding of the tree-based request structure, we provide a hypothetical requests network for an exemplar website (example.com), Figure 2. Each node represents a third-party receiving (and sometimes sending) a request, and each edge represents a request. The direction of the arrows shows the sender and receiver of a request. The size of each node implies the volume of data that is sent through the request body. The tree structure of the requests comprises different levels representing the flow of requests. In Figure 2, five out of twenty-one requests are directly sent to third-parties by the specific website (blue lines, level 1), while other requests are indirectly sent to other third-parties through the third-parties at different levels. For example, five requests are sent from third-parties at level 2 to third-parties at level 3 (green lines). Various

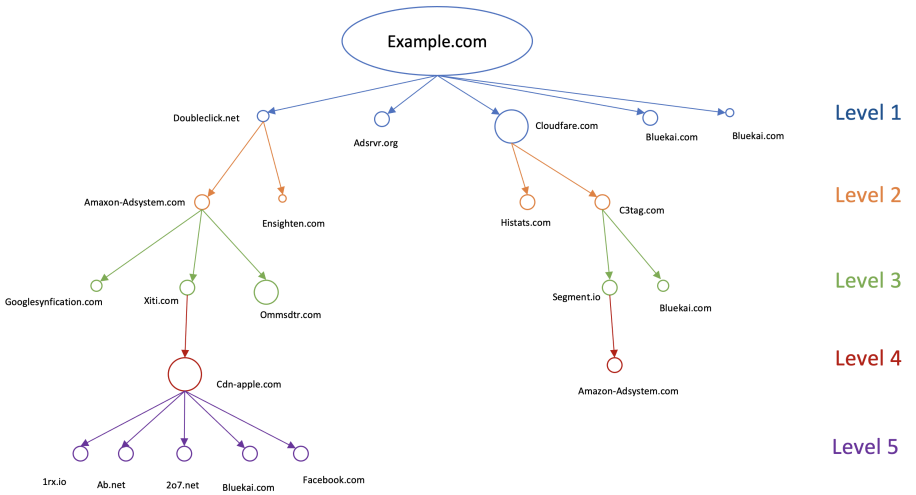

Figure 2: Sample Request Structure

websites have a different number of levels in their request structure. In the data set of this study, the maximum number of levels in request structures is 27. This illustrates how wide/long a request structure could be. Moreover, as each HTTP request is asking for a specific type of service, each third-party can appear multiple times at a level or in the entirety of the third-party request structure (e.g., Bluekai.com appears two times at level 1 and four times in the entire request structure).

In general, third-parties at level 1 are called third-parties or primary third-parties while other third-parties at other levels except one are called fourth-parties or secondary third-parties. The difference between primary third-parties and secondary third-parties is that primary third-parties receive requests originated from the website, while secondary third-parties receive requests sending from other third-parties rather than the website. In other words, secondary third-parties (or fourth-parties) are suppliers of the website suppliers. We collectively refer to all primary and secondary third-parties as third-parties in this study, and they are differentiated based on their level within the request struc-

tures in our data set.

4. Methodology

In this section, we will discuss how our prediction mechanism works to discern between legitimate and fraudulent websites. Our model consists of four steps including data collection, request structure extraction, variable extraction, and classification method. The following subsections describe each step.

4.1. Data Collection

The third-party request structure of our websites are collected. The goal is to differentiate between legitimate and fraudulent websites. We select legitimate websites from the list of top 50 websites provided by Alexa.com as of June 2020 (consistent with [54]) and end up with 32 websites from news category, 37 for education, 37 from health, 30 from science, 36 from e-commerce (online-electronic-markets), 23 from e-commerce (online-pharmacy markets), and 10 from online media.

For fraudulent websites, we use three sources. The first source is the "Counterfeit and Piracy Watch List" of The European Commission published in December 2018 [55]. The second source is the "2019 Review of Notorious Markets for Counterfeiting and Piracy" released by the Office of the U.S. Trade Representative in April of 2020 [56]. The third source of fraudulent websites is the list of fake news websites from Wikipedia.com[7]. We capture the third-party request structures and collect information on the third-parties for 205 legitimate and 93 fraudulent websites. Although we have an unbalanced sample size feeding into the prediction algorithms, results are validated by dividing the 205 legitimate websites into two randomly selected groups of 103 and 102. Each smaller group was then trained against the 93 fraudulent websites for a more balanced set. Results are approximately the same as for the unbalanced data, so we present the results for the single unbalanced data set.

Independent variables used in a prior study involving the use of third-parties for fraudulent website detection [4] were the ratio of the third-parties interacting with fraudulent and legitimate websites in the various classification clusters. These clusters were derived based on discovered relationships within the training set. In the current study, the value of the independent variables with respect to whether or not a third-party is known or safe is derived from independent sources. Thus, the independent variables in this study do not vary with the training set. The advantage of this approach is that results are more robust because they do not depend on having observed third-parties previously in the data collection. Rather, there is information about every third-party that is observed, even if that information is that a particular third-party is unknown to reputable internet resources on third-parties. The idea is that even the absence of information regarding a third-party is also information. Perhaps, the fact that a third-party is unknown to these reputable internet resources says quite a bit about whether or not a third-party has a propensity to do business with legitimate versus fraudulent websites. Thus, we collect data about third-parties in order to determine their characteristics. Using two sources, we are able to assign different attributes to third-parties in order to differentiate between them. The first data source about third-parties is the EasyPrivacy[1] list which defines whether a Third-party is safe or not. The second source is the Cookipedia[2] website which is a reliable and comprehensive data set providing detailed information about third-parties. Using this data we were able to determine whether a third-party is known to them or not.

In order to determine the business models underlying each third-party, we use four different sources: Cookipedia[2], Wappalyzer[3], Thirdpartytoday[4], and Webpagetest[5]. Based on the categorization of these aforementioned websites,

[1] https://easylist.to/easylist/easylist.txt
[2] https://cookiepedia.co.uk/
[3] https://www.wappalyzer.com/
[4] https://www.thirdpartyweb.today/
[5] https://Webpagetest.org/

Table 2: Number of third-parties based on the characteristic and business activity

		Characteristic				
		Safe	Unsafe	Known	Unknown	Total
Business Activity	Analytic	40	71	69	42	111
	Advertising	203	135	232	106	338
	Content Delivering	48	4	34	18	52
	Functionality	139	57	114	82	196
	Other	151	136	75	212	287
	Total	581	403	524	460	
	Total number of third-parties in the data set					984

we classify third-parties into five groups: Analytic, Advertising, Content Delivering, Functionality, and Others. Analytic third-parties mainly measure or track users and their actions on a website. Advertising third-parties are part of advertising and marketing networks, helping websites to deliver and manage advertising campaigns. Content Delivering third-parties, also called CDN (content delivery network), provide fast delivery of Internet content for websites. Functionality third-parties consist of those helping a website to continue to operate. This includes hosting platforms, developer utilities, and tag management. Several third-parties business activities were unknown, and thus we categorize them as Other. Table 2 shows the number of third-parties based on different characteristics and business activities. Categorizing third-parties based on these features (characteristics and business activity) help us to enhance our knowledge about them. This information is important in accurately classifying the website as fraudulent or legitimate.

4.2. Third-party Request Structure Extraction

Examination of HTML source code elements of the website misses many third-parties, and it does not provide details of the third-party request structure nor extended information about the third-parties. For this purpose, all third-parties in the request structure of a website are captured using Selenium WebDriver classes with the Python programming language. After capturing all

HTTP requests for a period of 15 seconds, we then identified the source and destination of each request which allows us to form the tree-based structure of all requests sent to various third-parties, as is visualized in Figure 1.

4.3. Variable Extraction

We capture the characteristics of the third-party request structure described in section 3 and information embedded in this structure with a variety of variables described in Table 3. The dependent variable of this study is *Website-Fakeness* which is a {0,1} binary categorical variable, whose value is 1 if the website is fraudulent, and 0 if the website is legitimate. Independent variables can be separated into two groups. First, variables related to the request structure alone include *NumberRequests*, *Number3P*, *DataSize*, $NumberRequests_i$, $Number3P_i$, $DataSize_i$, and *Depth*. These counting variables compute the number of requests, third-parties, and size of all requests' body in bytes in the whole request structure or in a specific level i respectively. For example, variable *NumberRequests* counts the total number of requests in the whole request structure of a website and $NumberRequests_i$ counts the number of requests where the sender of the request is located in the level *i-1* and the receiver is located in the level *i*. Using these variables we are able to capture quantitative characteristics of request structure. Table 3 summarizes variables, definitions and sources. The second group of variables relate to both third-party characteristics and the request structure, including $Number3P^x$ and $Number3P_i^x$ which express the number of the specific type of third-parties with feature x in the whole structure (total) and at level i respectively. Feature x includes whether a third-party is Safe (saf), UnSafe (uns), Known (kno), UnKnown (unk), Analytic (anl), Advertising (adv), Content Delivering (cnt), Functionality (fun), or Other (oth). For example $Number3P^{adv}$ measures the total number of advertising third-parties in the request structure and $Number3P_6^{adv}$ captures the number of advertising third-parties in level 6 of the request structure. The definition of each variable is provided in Table 3. A descriptive analysis of the request structure data is presented in Table 4.

Table 3: List of variables and definitions

Variables	Description	Source
Independent variables:		
Depth	The number of levels of the request structure of a given website.	Third-party Request Structure
Number3P	The total number of third-parties engaged in the request structure of a website.	Third-party Request Structure
Number3P$_i$	The number of third-parties in the level i of request structure.	Third-party Request Structure
NumberUnique3P	The total number of unique/non-duplicate third-parties in the request structure.	Third-party Request Structure
NumberRequests	The total number of requests in the request structure of a website.	Third-party Request Structure
NumberRequests$_i$	The number of requests in the level i of a request structure.	Third-party Request Structure
DataSize	The total size of all requests' body in bytes which are sent to third-parties in the whole request structure.	Third-party Request Structure
DataSize$_i$	The total size of requests' body in the level i of a request structure.	Third-party Request Structure
Number3P^x	The total number of third-parties with attribute x in the whole request structure.	Third-party Request Structure
	Attribute x:	
	- Safe(saf), UnSafe(uns)	EasyPrivacy
	- Known(kno), UnKnown(unk)	Cookipedia
	-Advertising(adv), Analytic(anl), Content delivering(cnt), Functionality(fun), and Other(oth)	Cookipedia, Wappalyzer, Thirdpartytoday, and Webpagetest
Number3P$_i^x$	The number of third-parties with attribute x in level i.	Third-party Request Structure
Dependent variable:		
WebsiteFakeness	1 if the website is fraudulent, and 0 if the website is legitimate.	

4.4. Classification Method

We used 17 different classification methods to predict whether a website is legitimate versus fraudulent. Sixteen of these classification methods represent a

Table 4: Descriptive Analysis of Data

Variables	Mean	Median	Minimum	Maximum
Depth	5.38	4	1	27
NumberRequests	89.12	50.5	1	620
DataSize	1448313.78	735729.5	43	35655268
Number3P	89.12	50.5	1	620
NumberUnique3P	19.93	13.5	1	92
Number3P^{saf}	68.07	38.5	0	566
Number3P^{uns}	18.05	4.5	0	166
Number3P^{kno}	74.37	36.5	0	560
Number3P^{unk}	11.75	4	0	200
Number3P^{anl}	9.32	5	0	98
Number3P^{adv}	48.15	14	0	548
Number3P^{cnt}	7.65	4	0	69
Number3P^{oth}	9.47	1	0	501
Number3P^{fun}	4.54	6	0	317

large cross-section of 14 available machine learning classification types presented in [57] as shown in Table A.11 in the Appendix. The seventeenth method, *CS5*, is a best-of-breed similarity measure developed by [4] to identify fake news websites. One of the primary goals of this study is to show that request structure can play a significant role in distinguishing between fraudulent versus legitimate websites. The results show that regardless of what type of predictive machine learning classification model is used, request structure data can be effectively used to detect fraudulent websites.

5. Computational Results

5.1. Neutral Baseline for Comparison

The main goal of this research is not necessarily to find "the best" classification method, but rather to show that incorporating the third-party request structure into predictive fraud models is a valuable contribution to the literature. Another goal of this paper is to begin to create classification techniques that work across a wide variety of website categories. To create a neutral base-

line, three naive prediction methods are used for benchmark comparisons: 1) predicting all websites as legitimate, 2) predicting all websites as fraudulent, and 3) a 50/50 coin toss for every website. Results for the three neutral baseline methods are presented in Table 5.

Table 5: Baseline Results

Method	Accuracy	Sensitivity	Specificity	Precision	F1	Matthews correlation coefficient	Youden J statistic
Naive Benchmark Models:							
All legitimate	0.688	1.000	0.000	1.000	0.500	0.000	0.000
All fraudulent	0.312	0.480	1.000	0.000	0.310	0.000	0.000
50/50 coin flip	0.500	0.380	0.500	0.500	0.310	0.000	0.000
Predictive Methods Using Third-party Usage (3PU) Data:							
$CS5_{3PU}$	0.728	0.790	0.591	0.810	0.800	0.376	0.382
$svmRadial_{3PU}$	0.721	0.927	0.266	0.738	0.821	0.291	0.192
mlp_{3PU}	0.715	0.888	0.330	0.747	0.810	0.266	0.218
knn_{3PU}	0.711	0.864	0.372	0.754	0.804	0.272	0.236
lvq_{3PU}	0.710	0.926	0.232	0.730	0.815	0.230	0.158
rf_{3PU}	0.707	0.824	0.447	0.771	0.795	0.291	0.271
$gausspRadial_{3PU}$	0.704	0.903	0.266	0.732	0.807	0.222	0.168
$rpart_{3PU}$	0.701	0.859	0.351	0.746	0.797	0.244	0.210
fda_{3PU}	0.698	0.903	0.242	0.727	0.804	0.197	0.145
$svmLinear_{3PU}$	0.688	1.000	0.000	0.688	0.815	0.000	0.000
$glmnet_{3PU}$	0.688	1.000	0.000	0.688	0.815	0.000	0.000
$gausspPoly_{3PU}$	0.688	1.000	0.000	0.688	0.815	0.000	0.000
plr_{3PU}	0.688	1.000	0.000	0.688	0.815	0.000	0.000
pls_{3PU}	0.688	1.000	0.000	0.688	0.815	0.000	0.000
sda_{3PU}	0.684	0.966	0.063	0.695	0.808	0.069	0.029
ada_{3PU}	NaN	NaN	NaN	NaN	NaN	NaN	NaN
$glmboost_{3PU}$	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Note 1: Full method names are shown in Table A.11.
Note 2: 3PU data is first utilized in [4].
Note 3: Methods $CS5_{3PU}$ and $svmLinear_{3PU}$ are first utilized in [4].

5.2. Baseline Methods

Gopal et al. [4] use the third-party relationships of a website to detect fake news and clickbait websites. Since the data model they use (third-party usage (3PU)) can also be constructed from our data set, we first perform all 17 predictive models utilizing their third-party usage (3PU) similarity data structure. This allows us to determine how well the 3PU data structure method works for a variety of methods when we extend the type of websites from news (which are the only type of websites used in [4]) to a wider variety of websites types. The

Table 6: Results Using Third-party Request Structure (RS) Data

Method	Accuracy	Sensitivity	Specificity	Precision	F1	Matthews correlation coefficient	Youden J statistic
Predictive Methods Using Third-party Request Structure (RS) Data:							
svmRadial$_{RS}$	0.771	0.893	0.500	0.801	0.843	0.437	0.393
ada$_{RS}$	0.761	0.888	0.479	0.793	0.837	0.410	0.367
CS5$_{RS}$	0.758	0.868	0.516	0.798	0.832	0.410	0.384
rf$_{RS}$	0.755	0.814	0.621	0.824	0.816	0.435	0.435
mlp$_{RS}$	0.752	0.830	0.578	0.816	0.821	0.415	0.408
lvq$_{RS}$	0.752	0.874	0.479	0.791	0.826	0.388	0.352
knn$_{RS}$	0.742	0.834	0.537	0.800	0.815	0.383	0.371
glmnet$_{RS}$	0.725	0.893	0.350	0.754	0.816	0.296	0.243
rpart$_{RS}$	0.721	0.815	0.512	0.789	0.797	0.338	0.327
gausspRadial$_{RS}$	0.720	0.863	0.401	0.764	0.806	0.304	0.264
fda$_{RS}$	0.719	0.806	0.524	0.787	0.794	0.336	0.330
glmboost$_{RS}$	0.718	0.840	0.446	0.772	0.802	0.310	0.285
gausspPoly$_{RS}$	0.711	0.930	0.225	0.728	0.816	0.226	0.154
plr$_{RS}$	0.708	0.762	0.588	0.805	0.778	0.343	0.350
sda$_{RS}$	0.708	0.830	0.436	0.765	0.793	0.287	0.266
svmLinear$_{RS}$	0.698	0.747	0.588	0.801	0.771	0.326	0.335
pls$_{RS}$	0.695	0.922	0.192	0.716	0.806	0.168	0.115

Note 1: Full method names are shown in Table A.11.

results are presented in Table 5.The 10-fold cross-validation technique is used for all machine learning methods.

5.3. Results Using Request Structure Data and Ensemble Improvements

We next apply the machine learning methods to the third-party request structure (RS) data model (used by this paper). Results using the RS data presented in Table 6 show a substantial improvement over the baseline results using $3PU$ data. Results of the predictive models using combined third-party usage (3PU) and third-party request structure (RS) data is presented in Table 7. Many of the methods show improved results when using combined data. A paired t-test is conducted to compare the differences between different data models, and the p-value results for each of the performance metrics are presented in Table 8. We observe that both RS and the combined $3PU, RS$ data models are significantly better than $3PU$ for all performance metrics except the overall F1 score. The Matthews correlation coefficient and Youden J statistic, which are also overall measures, do show significant improvement across the methods. Note that individual methods do show improvements in the F1 score. A variety

of ensemble methods are presented in Appendix A in Table A.12. Ensembles include best of three, best of five, best of nine, and weighted. Weighted ensemble methods perform best, yielding substantial improvements on all measures.

Table 7: Predictive Model Results Using Combined Third-party Usage (3PU) and Third-party Request Structure (RS) Data

Method	Accuracy	Sensitivity	Specificity	Precision	F1	Matthews correlation coefficient	Youden J statistic
Predictive Methods Using Third-party Usage (3PU) and Third-party Request Structure (RS) Data							
$CS5_{3PU,RS}$	0.805	0.868	0.667	0.852	0.860	0.542	0.535
$svmRadial_{3PU,RS}$	0.768	0.888	0.506	0.803	0.841	0.430	0.394
$rf_{3PU,RS}$	0.744	0.878	0.447	0.778	0.824	0.366	0.325
$lvq_{3PU,RS}$	0.735	0.835	0.511	0.798	0.812	0.363	0.346
$knn_{3PU,RS}$	0.735	0.845	0.490	0.791	0.813	0.357	0.335
$fda_{3PU,RS}$	0.735	0.815	0.557	0.800	0.806	0.377	0.372
$glmnet_{3PU,RS}$	0.729	0.825	0.512	0.791	0.803	0.351	0.337
$mlp_{3PU,RS}$	0.728	0.800	0.567	0.807	0.799	0.369	0.367
$rpart_{3PU,RS}$	0.721	0.815	0.512	0.789	0.797	0.338	0.327
$gausssprRadial_{3PU,RS}$	0.720	0.893	0.336	0.752	0.814	0.285	0.228
$svmLinear_{3PU,RS}$	0.702	0.747	0.598	0.804	0.770	0.336	0.345
$gausssprPoly_{3PU,RS}$	0.692	1.000	0.011	0.691	0.817	0.086	0.011
$pls_{3PU,RS}$	0.688	1.000	0.000	0.688	0.815	0.000	0.000
$plr_{3PU,RS}$	0.688	0.733	0.588	0.798	0.759	0.310	0.320
$sda_{3PU,RS}$	0.684	0.810	0.403	0.751	0.775	0.231	0.214
$glmboost_{3PU,RS}$	NaN	NaN	NaN	NaN	NaN	NaN	NaN
$ada_{3PU,RS}$	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Note 1: Full method names are shown in Table A.11.

Table 8: P-values for Paired T-tests Comparing Classification Performance

Data Model	Accuracy	Sensitivity	Specificity	Precision	F1	Matthews correlation coefficient	Youden J statistic
RS versus 3PU	<0.001*	0.002*	<0.001*	<0.001*	0.942	<0.001*	<0.001*
Combined (3PU,RS) versus 3PU	0.019*	0.011*	0.002*	0.001*	0.771	0.004*	0.002*
Combined (3PU,RS) versus RS	0.701	0.845	0.710	0.826	0.844	0.556	0.677

*P-values significant at alpha = 0.05

5.4. Comparison of 3PU versus RS Data Using CS5 and svmRadial Methods

We next take two methods that perform well, $CS5$ and *svmRadial*, and compare the results as we apply them to the existing, proposed, and combined data models, as well as an ensemble model. Among the various methods to detect fraudulent news websites reported in Gopal et al. [4] based on third-party usage, two of the best are: 1) Cosine similarity classifier ($CS5$), and 2) Machine learning method (*svmLinear*). We will use *svmRadial* instead of *svmLinear*

for our comparison due to superior results with this data. As illustrated in Table 9, we observe substantial improvements in performance metrics across the board. The ensemble method yields the best results compared to all methods, improving accuracy from 0.728 and 0.721 using $3PU$ data and RS data respectively to 0.826. The F1 score improves from 0.800 and 0.821 to 0.876.

Table 9: Performance of the Existing, Proposed, and Ensemble Methods Using Third-party Usage ($3PU$) and Third-party Request Structure (RS) for $CS5$ and *svmRadial* Methods

Method	Accuracy	Sensitivity	Specificity	Precision	F1	Matthews correlation coefficient	Youden J statistic
Existing Methods Using Third-party Usage:							
$CS5_{3PU}$[4]	0.728	0.790	0.591	0.810	0.800	0.376	0.382
$svmRadial_{3PU}$	0.721	0.927	0.266	0.738	0.821	0.266	0.192
Proposed Methods Using Third-party Request Structure:							
$CS5_{RS}$	0.758	0.868	0.516	0.798	0.832	0.410	0.384
$svmRadial_{RS}$	0.771	0.893	0.500	0.801	0.843	0.437	0.393
Combined Data Model Using Third-party Usage and Third-party Request Structures:							
$CS5_{3PU,RS}$	0.805	0.868	0.667	0.852	0.860	0.542	0.535
$svmRadial_{3PU,RS}$	0.768	0.888	0.506	0.803	0.841	0.430	0.394
Proposed Ensemble Model:							
Ensemble Model*	0.826	0.893	0.677	0.859	0.876	0.585	0.570

*Ensemble method is across $CS5_{3PU}$[4], $svmRadial_{3PU}$, $CS5_{RS}$, $svmRadial_{RS}$, $CS5_{3PU,RS}$, and $svmRadial_{3PU,RS}$, with double emphasis on $CS5_{3PU,RS}$ and $svmRadial_{RS}$ which are the two best of these six methods.

5.5. Feature Importance Analysis

After removing variables with zero and near zero variance, 151 variables remain in the dataset. We use three methods to examine feature importance. Two methods are based on prediction models: Support Vector Machine (svm) and Random Forest (rf). One method is the correlation between the dependent variable (*WebsiteFakeness*) and the independent variables. The importance score for each variable is calculated and then scaled from 0 to 100. A variable with higher importance score is considered as the more important variable used in detection of fraudulent websites. Table 10 shows the top 20 variables with the highest importance score for each of the three methods. Among the top 20 most important variables for each method, we observe that the total number of analytic third-parties ($Number3P^{anl}$) is the most important variable in all three methods. Moreover, the results of feature importance analysis show that features related to different levels of the requests structure of a website are very

useful in predicting whether a website is fraudulent or legitimate. This analysis also supports our claim that request structure features, along with features related to third-parties, are useful for the identification of fraudulent websites.

6. Conclusion

Utilizing the structure of the third-party request and publicly available attributes related to the third-parties, this research strongly supports the assertion that the request structure of websites can be used to create effective and efficient prediction models to distinguish between legitimate and fraudulent websites across a variety of website categories. We showed that the number of third-parties employed, their location in the website request structure, attributes such as safe/unsafe and known/unknown of third-parties, and business models of third-parties are contributing significantly to the prediction of fraudulent websites. A wide variety of predictive models that can operate in real-time perform well in distinguishing between legitimate and fraudulent websites. This shows that the effectiveness of incorporating the third-party request structure is robust across a wide variety of prediction models.

6.1. Practical Implications

Given that fraudulent websites come and go as they are identified, the window that is available for doing damage, while perhaps small, can provide opportunity to do significant economic harm. Therefore, any mechanism to detect fraudulent websites in real-time can be most useful. Real-time decision support for detection of fraudulent website before damage is done is worthy of future study. With billions of dollars of fraudulent revenue, collective efforts and actions are required to fight against Internet fraud [5, 58]. The approach in this paper is meant to complement conventional detection methods as it is more generalized, less costly, less computationally complex, and less time-consuming in comparison to existing methods. While content-based approaches have the weakness that fraudulent websites can alter their appearance to avoid being

detected, our approach overcomes this flaw as request structures are difficult to hide and manipulate. The request structures reflect the underlying business models employed by websites and third-parties. It is costly in time, effort and expense for a fraudulent website to adopt the request structure of a legitimate website. There are good reasons why the request structures differ.

Additionally, our method illustrates that it is possible to improve upon the benchmark and existing prediction method results across a wide category of fraudulent websites (e.g, news, health, e-commerce, online media, etc.). Our analysis shows that in different categories, the models are able to predict fraudulent websites well. Not only is this good news for users, but it is also good news for credit card companies and e-commerce platforms that are concerned about legitimate online transactions and products.

6.2. Theoretical Implications

We additionally contribute to literature by defining and identifying request structures and relevant attributes. Third-party request structure can be useful in other research topics such as security and privacy. It is important to note that websites are likely not fully aware of every single third-party that is involved with their website, as the third-parties can vary from visit to visit and many calls are deeply embedded in the lower levels of the request structure. Website users are also likely unaware of all of this activity as well.

This study opens many theoretical and research possibilities with respect to the use of website third-party request structure, beyond simply identifying fraudulent websites. This paper also serves to enlighten policy makers and regulators as to the nature of website utilization of third-parties in request structures, how third-parties collaborate with websites and other third-parties, and how relevant request structure characteristics can be extracted.

6.3. Limitations and Future Research

One of the limitations of this study is the size of the data set. Future research should work to increase the sample size. The problem is that fraudulent

websites change URLs rather quickly once discovered. The current work could be used to identify additional fraudulent websites in the wild. Moreover, although we showed that the request structures can be used to classify the true nature of websites in terms of being fraudulent or legitimate, future research is needed to improve the accuracy of generalized algorithms that apply across a wide variety of website categories. Further, analysis of the underlying economics driving request structures may shed more important light on the business relationships within websites and third-parties eco-systems. Special structures in various categories such as specific industries, national versus global markets, and cyber versus physical networks potentially have attributes that can be used to drill down and identify anomalies that can help detect fraudulent websites more precisely. Data from blockchains, social networks, and Internet of Things (IoT) have the potential to expose opportunities to identify these special structures.

References

[1] P. Nohe, Re-hashed: 2018 cybercrime statistics: A closer look at the "web of profit", Available at *www.thesslstore.com/blog/2018-cybercrime-statistics/*, 2018.

[2] N. Statt, Amazon will start listing names and addresses of marketplace sellers to combat counterfeiting, Available at *www.theverge.com/2020/7/8/21317617/amazon-counterfeit-products-marketplace-sellers-names-addresses-transparency/*, 2020.

[3] C. Crane, Phishing statistics: The 29 latest phishing stats to know in 2020., Available at *www.thesslstore.com/blog/phishing-statistics-latest-phishing-stats-to-know/*, 2020.

[4] R. D. Gopal, H. Hidaji, S. N. Kutlu, R. A. Patterson, E. Rolland, D. Zhdanov, Real or not? identifying untrustworthy news websites using third-

Table 10: Feature Importance Analysis (Top 20 Features)

	Correlation with Dependant Variable		**Using Support Vector Machine (SVM)**		**Using Random Forest**	
	Variables	**Score**	**Variables**	**Score**	**Variables**	**Score**
1	$Number3P^{anl}$	100.0	$Number3P^{anl}$	100.0	$Number3P^{anl}$	100.0
2	$Number3P_{3}^{anl}$	99.1	$Number3P^{uns}$	93.7	$Number3P_{6}^{adv}$	97.8
3	$NumberUnique3P$	95.8	$Number3P_{3}^{anl}$	91.1	$Number3P_{2}^{cnt}$	96.9
4	$Number3P_{2}^{kno}$	88.3	$NumberUnique3P$	87.3	$NumberRequests_{10}$	92.0
5	$Number3P_{3}^{kno}$	86.9	$Number3P_{2}^{adv}$	85.8	$Number3P^{oth}$	91.2
6	$Number3P_{2}^{uns}$	85.7	$Number3P_{3}^{kno}$	81.5	$Number3P_{11}^{kno}$	87.3
7	$Number3P_{3}^{uns}$	84.9	$Number3P_{2}^{uns}$	81.2	$NumberRequests_{13}$	87.2
8	$Number3P_{4}^{anl}$	82.5	$Number3P_{2}^{kno}$	80.6	$Number3P_{4}^{oth}$	86.4
9	$Number3P_{2}^{saf}$	81.9	$Number3P^{adv}$	78.2	$Number3P_{12}^{kno}$	86.0
10	$Number3P^{uns}$	81.4	$Number3P_{2}^{anl}$	76.7	$Number3P^{cnt}$	84.7
11	$Number3P_{3}^{saf}$	78.8	$Number3P_{3}^{saf}$	76.5	$Number3P_{3}^{cnt}$	84.4
12	$Number3P_{1}^{uns}$	76.4	$Number3P^{kno}$	74.3	$Number3P_{12}^{uns}$	83.8
13	$Number3P_{1}^{kno}$	75.8	$Number3P_{2}^{saf}$	71.9	$NumberRequests_{11}$	82.8
14	$Number3P_{2}^{adv}$	74.1	$Number3P^{unk}$	69.2	$Number3P_{10}^{kno}$	82.6
15	$Number3P_{4}^{saf}$	71.7	$NumberRequest$	69.2	$Number3P_{11}^{uns}$	82.5
16	$Number3P_{1}^{saf}$	70.2	$Number3P$	69.2	$Number3P_{13}^{kno}$	82.2
17	$Number3P^{adv}$	68.9	$NumberRequests_{3}$	68.8	$NumberRequests_{12}$	81.9
18	$Number3P_{4}^{kno}$	66.1	$DataSize_{3}$	68.5	$Number3P_{13}^{uns}$	79.3
19	$Number3P_{3}^{adv}$	65.5	$NumberRequests_{2}$	68.5	$Number3P_{10}^{adv}$	78.2
20	$Number3P_{2}^{unk}$	64.3	$Number3P_{1}^{uns}$	67.4	$Number3P_{3}^{anl}$	77.2

party partnerships, ACM Transactions on Management Information Systems (TMIS) 11 (2020) 1–20.

[5] C. E. H. Chua, J. Wareham, D. Robey, The role of online trading communities in managing internet auction fraud, MIS Quarterly (2007) 759–781.

[6] Y. Zhang, S. Egelman, L. Cranor, J. Hong, Phinding phish: Evaluating anti-phishing tools (2007).

[7] Wikipedia contributors, List of fake news websites - Wikipedia, the free encyclopedia, 2020. URL: *https://en.wikipedia.org/wiki/List_of_fake_news_websites*, [Online; accessed 16-June-2020].

[8] HarvardLibrary, Research guides: Fake news, misinformation, and propaganda., *https://guides.library.harvard.edu/fake*, 2017.

[9] A. Abbasi, F. ahedi, D. Zeng, Y. Chen, H. Chen, J. F. Nunamaker Jr, Enhancing predictive analytics for anti-phishing by exploiting website genre information, Journal of Management Information Systems 31 (2015) 109–157.

[10] N. C. R. L. Y. Teraguchi, J. C. Mitchell, Client-side defense against web-based identity theft, Computer Science Department, Stanford University. Available: http://crypto. stanford. edu/SpoofGuard/webspoof. pdf (2004).

[11] C. L. Tan, K. L. Chiew, K. Wong, et al., Phishwho: Phishing webpage detection via identity keywords extraction and target domain name finder, Decision Support Systems 88 (2016) 18–27.

[12] A. C. Bahnsen, E. C. Bohorquez, S. Villegas, J. Vargas, F. A. González, Classifying phishing urls using recurrent neural networks, in: 2017 APWG symposium on electronic crime research (eCrime), IEEE, 2017, pp. 1–8.

[13] A. A. Ubing, S. K. B. Jasmi, A. Abdullah, N. Jhanjhi, M. Supramaniam, Phishing website detection: An improved accuracy through feature selec-

tion and ensemble learning, International Journal of Advanced Computer Science and Applications 10 (2019) 252–257.

[14] Y. Zhang, J. I. Hong, L. F. Cranor, Cantina: a content-based approach to detecting phishing web sites, in: Proceedings of the 16th international conference on World Wide Web, 2007, pp. 639–648.

[15] W. Liu, X. Deng, G. Huang, A. Y. Fu, An antiphishing strategy based on visual similarity assessment, IEEE Internet Computing 10 (2006) 58–65.

[16] M. Wu, R. C. Miller, S. L. Garfinkel, Do security toolbars actually prevent phishing attacks?, in: Proceedings of the SIGCHI conference on Human Factors in computing systems, 2006, pp. 601–610.

[17] G. Xiang, J. Hong, C. P. Rose, L. Cranor, Cantina+ a feature-rich machine learning framework for detecting phishing web sites, ACM Transactions on Information and System Security (TISSEC) 14 (2011) 1–28.

[18] A. Abbasi, Z. Zhang, D. Zimbra, H. Chen, J. F. Nunamaker Jr, Detecting fake websites: The contribution of statistical learning theory, Mis Quarterly (2010) 435–461.

[19] A. Y. Fu, L. Wenyin, X. Deng, Detecting phishing web pages with visual similarity assessment based on earth mover's distance (emd), IEEE transactions on dependable and secure computing 3 (2006) 301–311.

[20] H. Zhang, G. Liu, T. W. Chow, W. Liu, Textual and visual content-based anti-phishing: a bayesian approach, IEEE transactions on neural networks 22 (2011) 1532–1546.

[21] K. L. Chiew, E. H. Chang, W. K. Tiong, et al., Utilisation of website logo for phishing detection, Computers & Security 54 (2015) 16–26.

[22] L. Wenyin, N. Fang, X. Quan, B. Qiu, G. Liu, Discovering phishing target based on semantic link network, Future Generation Computer Systems 26 (2010) 381–388.

[23] H. Yuan, X. Chen, Y. Li, Z. Yang, W. Liu, Detecting phishing websites and targets based on urls and webpage links, in: 2018 24th International Conference on Pattern Recognition (ICPR), IEEE, 2018, pp. 3669–3674.

[24] S. Marchal, K. Saari, N. Singh, N. Asokan, Know your phish: Novel techniques for detecting phishing sites and their targets, in: 2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS), IEEE, 2016, pp. 323–333.

[25] S. Smadi, N. Aslam, L. Zhang, Detection of online phishing email using dynamic evolving neural network based on reinforcement learning, Decision Support Systems 107 (2018) 88–102.

[26] P. Yang, G. Zhao, P. Zeng, Phishing website detection based on multidimensional features driven by deep learning, IEEE Access 7 (2019) 15196–15209.

[27] S. Y. Yerima, M. K. Alzaylaee, High accuracy phishing detection based on convolutional neural networks, in: 2020 3rd International Conference on Computer Applications & Information Security (ICCAIS), IEEE, 2020, pp. 1–6.

[28] J. Wadleigh, J. Drew, T. Moore, The e-commerce market for" lemons" identification and analysis of websites selling counterfeit goods, in: Proceedings of the 24th International Conference on World Wide Web, 2015, pp. 1188–1197.

[29] M. Maktabar, A. Zainal, M. A. Maarof, M. N. Kassim, Content based fraudulent website detection using supervised machine learning techniques, in: International Conference on Health Information Science, Springer, 2017, pp. 294–304.

[30] C. Carpineto, G. Romano, Learning to detect and measure fake ecommerce websites in search-engine results, in: Proceedings of the International Conference on Web Intelligence, 2017, pp. 403–410.

[31] W. Mostard, B. Zijlema, M. Wiering, Combining visual and contextual information for fraudulent online store classification, in: IEEE/WIC/ACM International Conference on Web Intelligence, 2019, pp. 84–90.

[32] M. N. Kassim, M. A. Maarof, M. Bakhtiari, Fraudulent e-commerce website detection model using html, text and image features, in: Proceedings of the 11th International Conference on Soft Computing and Pattern Recognition (SoCPaR 2019), volume 1182, Springer Nature, 2020, p. 177.

[33] L. Beltzung, A. Lindley, O. Dinica, N. Hermann, R. Lindner, Real-time detection of fake-shops through machine learning, in: 2020 IEEE International Conference on Big Data (Big Data), IEEE, 2020, pp. 2254–2263.

[34] C. Castillo, M. Mendoza, B. Poblete, Information credibility on twitter, in: Proceedings of the 20th international conference on World wide web, 2011, pp. 675–684.

[35] A. Gupta, P. Kumaraguru, C. Castillo, P. Meier, et al., Tweetcred: A real-time web-based system for assessing credibility of content on twitter, arXiv preprint arXiv:1405.5490 (2014).

[36] Y. Chen, N. J. Conroy, V. L. Rubin, Misleading online content: recognizing clickbait as" false news", in: Proceedings of the 2015 ACM on workshop on multimodal deception detection, 2015, pp. 15–19.

[37] H. Ahmed, I. Traore, S. Saad, Detection of online fake news using n-gram analysis and machine learning techniques, in: International conference on intelligent, secure, and dependable systems in distributed and cloud environments, Springer, 2017, pp. 127–138.

[38] P. M. Konkobo, R. Zhang, S. Huang, T. T. Minoungou, J. A. Ouedraogo, L. Li, A deep learning model for early detection of fake news on social media, in: 2020 7th International Conference on Behavioural and Social Computing (BESC), IEEE, 2020, pp. 1–6.

[39] K. Chow, K. Cheng, L. Man, P. K. Lai, L. C. Hui, C. Chong, K. Pun, W. Tsang, H. Chan, S. Yiu, Btm-an automated rule-based bt monitoring system for piracy detection, in: Second International Conference on Internet Monitoring and Protection (ICIMP 2007), IEEE, 2007, pp. 2–2.

[40] B. Srinivas, K. V. Rao, P. S. Varma, Movie piracy detection based on audio features using mel-frequency cepstral coefficients and vector quantization, International Journal of Soft Computing and Engineering 2 (2012) 27–31.

[41] C. S. Gosavi, S. N. Mali, Video authentication and copyright protection using unique watermark generation technique and singular value decomposition, International Journal of Computer Applications 123 (2015).

[42] A. Tiwari, H. Shah, U. Thube, N. Shah, Video piracy detection using invisible watermark, International Journal of Engineering Research Technology (IJERT) (2015).

[43] A. M. Chavan, Piracy detection of video contents by signature matching method, Int. J. Comput. Eng. Res. Trends 5 (2018) 202–206.

[44] N. Astiqah Omar, Z. Z. M. Zakuan, R. Saian, Software piracy detection model using ant colony optimization algorithm, in: Journal of Physics Conference Series, volume 855, 2017, p. 012031.

[45] S. Kazi, M. Stamp, Hidden markov models for software piracy detection, Information Security Journal: A Global Perspective 22 (2013) 140–149.

[46] R. Yasaei, S.-Y. Yu, E. K. Naeini, M. A. A. Faruque, Gnn4ip: Graph neural network for hardware intellectual property piracy detection, arXiv preprint arXiv:2107.09130 (2021).

[47] K. Rudman, M. Bonenfant, M. Celik, J. Daniel, J. Haitsma, J.-P. Panis, Toward real-time detection of forensic watermarks to combat piracy by live streaming, in: SMPTE 2014 Annual Technical Conference & Exhibition, SMPTE, 2014, pp. 1–12.

[48] H. Sudler, Effectiveness of anti-piracy technology: Finding appropriate solutions for evolving online piracy, Business Horizons 56 (2013) 149–157.

[49] S.-K. Choi, J. Kwak, Feature analysis and detection techniques for piracy sites, KSII Transactions on Internet and Information Systems (TIIS) 14 (2020) 2204–2220.

[50] E.-J. Kim, J. Kwak, Intelligent piracy site detection technique with high accuracy, KSII Transactions on Internet and Information Systems (TIIS) 15 (2021) 285–301.

[51] A. Le, A. Markopoulou, M. Faloutsos, Phishdef: Url names say it all, in: 2011 Proceedings IEEE INFOCOM, IEEE, 2011, pp. 191–195.

[52] J. C. Reis, A. Correia, F. Murai, A. Veloso, F. Benevenuto, Supervised learning for fake news detection, IEEE Intelligent Systems 34 (2019) 76–81.

[53] K. Shu, S. Wang, H. Liu, Beyond news contents: The role of social context for fake news detection, in: Proceedings of the twelfth ACM international conference on web search and data mining, 2019, pp. 312–320.

[54] R. D. Gopal, H. Hidaji, R. A. Patterson, E. Rolland, D. Zhdanov, How much to share with third parties? user privacy concerns and website dilemmas, MIS Quarterly 42 (2018) 143–164.

[55] EuropeanCommission, Counterfeit and piracy watch list, Available at *https://trade.ec.europa.eu/doclib/docs/2018/december/tradoc-157564.pdf*, 2018.

[56] USTR, The office of the united states trade representative, 2019 review of notorious markets for counterfeiting and piracy, Available at *https://ustr.gov/sites/default/files/2019-Review-of-Notorious-Markets-for-Counterfeiting-and-Piracy.pdf*, 2020.

[57] R Docimentation, train: Fit predictive models over different tuning parameters, 2021. URL: *https://www.rdocumentation.org/packages/caret/versions/5.16-24/topics/train*, [Online; accessed 25-Sep-2021].

[58] T. Dinev, Why spoofing is serious internet fraud, Communications of the ACM (2006) 76–82.

Appendix A.

Table A.11: Predictive Methods Used In This Study

	Abbreviation	Method Full Name	Classification Type
1	ada	Boosted Classification Trees	Boosted Trees
2	fda	Flexible Discriminant Analysis	Discriminant Analysis
3	gaussprPoly	Gaussian Processes with Polynomial kernel	Gaussian Processes
4	gaussprRadial	Gaussian Process with Radial Kernel	Gaussian Processes
5	glmboost	Boosted Generalized Linear Model	Boosting (Non-Tree)
6	glmnet	Elastic-Net Regularized Generalized Linear Mode	Elastic Net
7	knn	k-Nearest Neighbors	K Nearest Neighbor
8	lvq	Learning Vector Quantization	Learned Vector Quantization
9	mlp	Multi-Layer Perception Neural Network	Neural Networks
10	plr	Penalized Logistic Regression	Logistic/Multinomial Regression
11	pls	Partial Least Squares	Partial Least Squares
12	rf	Random Forest	Random Forest
13	rpart	Recursive Partitioning and Regression Trees	Recursive Partitioning
14	sda	Shrinkage Discriminant Analysis	Linear Discriminant Analysis
15	svmLinear	Support Vector Machines with Linear kernel	Support Vector Machines
16	svmRadial	Support Vector Machines with Radial kernel	Support Vector Machines
17	CS5	Cosine Similarity	Similarity heuristic [4]

Note 1: Unless otherwise noted, methods are selected from the "Caret" package in R [57].

Table A.12: Results of Various Ensemble Models

Row	Method	Accuracy	Sensitivity	Specificity	Precision	F1	Matthews correlation coefficient	Youden J statistic
Best of Three Ensemble Models:								
1	Top 3 ML methods from Combination Table ($CS5_{3PU,RS}, svmRadial_{3PU,RS}, rf_{3PU,RS}$)	0.728	0.932	0.280	0.740	0.825	0.287	0.211
2	Top 3 ML methods from RS Table (svmRadial$_{RS}, ada_{RS}, CS5_{RS}$)	0.779	0.907	0.495	0.798	0.849	0.451	0.402
3	Top 3 ML methods from 3PU Table (knn$_{3PU}, CS5_{3PU}, svmRadial_{3PU}$)	0.779	0.902	0.505	0.801	0.849	0.453	0.408
4	Top 3 overall from 3PU, RS and combination table ($CS5_{3PU,RS}, CS5_{3PU}, svmRadial_{3PU}$)	0.802	0.848	0.693	0.868	0.858	0.532	0.541
5	Top 3 overall from 3PU, RS, and combination table without duplication of ML method ($CS5_{3PU,RS}, svmRadial_{RS}, ada_{RS}$)	0.775	0.912	0.473	0.792	0.848	0.440	0.385
Best of Five Ensemble Models:								
6	Top 5 overall from 3PU, RS, and Combination table ($CS5_{3PU,RS}, svmRadial_{3PU,RS}$, svmRadial$_{RS}, ada_{RS}, CS5_{RS}$)	0.789	0.912	0.516	0.806	0.856	0.478	0.428
7	Top 5 overall from 3PU, RS, and Combination table without duplication of ML method ($CS5_{3PU,RS}, svmRadial_{RS}, ada_{RS}, rf_{RS}, lvq_{Rs}$)	0.772	0.907	0.473	0.791	0.845	0.432	0.380
Best of Nine Ensemble Model:								
8	Combined Top 3 from every table (best of 9) ($CS5_{3PU,RS}, CS5_{3PU}, svmRadial_{3PU}$, svmRadial$_{RS}, ada_{RS}, CS5_{RS}$, $CS5_{3PU,RS}, svmRadial_{3PU,RS}, rf_{3PU,RS}$)	0.765	0.917	0.430	0.780	0.843	0.409	0.347
Weighted Ensemble Models:								
9	Weighted Ensemble Model across CS5$_{3PU}$[4], $svmRadial_{3PU}, CS5_{RS}$, svmRadial$_{RS}, CS5_{3PU,RS}, svmRadial_{3PU,RS}$ with double emphasis on $CS5_{3PU,RS}, svmRadial_{RS}$	0.826	0.893	0.677	0.859	0.876	0.585	0.570
10	Weighted Ensemble Model of $CS5_{RS}, svmRadial_{RS}, CS5_{3PU,RS}, svmRadial_{3PU,RS}$ with double emphasis on $CS5_{3PU,RS}, svmRadial_{3PU,RS}$	0.826	0.888	0.688	0.863	0.875	0.587	0.576

Note 1: Full method names are shown in Table A.11.