# "*Kn0w Thy Doma1n Name*": Unbiased Phishing Detection Using Domain Name Based Features

Hossein Shirazi
Computer Science Department
Colorado State University
Fort Collins, Colorado, USA
shirazi@colostate.edu

Bruhadeshwar Bezawada
Computer Science Department
Colorado State University
Fort Collins, Colorado, USA
bru.bezawada@colostate.edu

Indrakshi Ray
Computer Science Department
Colorado State University
Fort Collins, Colorado, USA
indrakshi.ray@colostate.edu

## ABSTRACT

Phishing websites remain a persistent security threat. Thus far, machine learning approaches appear to have the best potential as defenses. But, there are two main concerns with existing machine learning approaches for phishing detection. The first is the large number of training features used and the lack of validating arguments for these feature choices. The second concern is the type of datasets used in the literature that are inadvertently biased with respect to the features based on the website URL or content. To address these concerns, we put forward the intuition that the *domain name* of phishing websites is *the tell-tale* sign of phishing and holds the key to successful phishing detection. Accordingly, we design features that model the relationships, visual as well as statistical, of the domain name to the key elements of a phishing website, which are used to snare the end-users. The main value of our feature design is that, to bypass detection, an attacker will find it very difficult to tamper with the visual content of the phishing website without arousing the suspicion of the end user. Our feature set ensures that there is minimal or no bias with respect to a dataset. Our learning model trains with only seven features and achieves a true positive rate of 98% and a classification accuracy of 97%, on sample dataset. Compared to the state-of-the-art work, our per data instance classification is 4 times faster for legitimate websites and 10 times faster for phishing websites. Importantly, we demonstrate the shortcomings of using features based on URLs as they are likely to be biased towards specific datasets. We show the robustness of our learning algorithm by testing on unknown live phishing URLs and achieve a high detection accuracy of 99.7%.

## KEYWORDS

Phishing, Phishing detection, Domain name, Machine Learning, Biased datasets

**ACM Reference Format:**
Hossein Shirazi, Bruhadeshwar Bezawada, and Indrakshi Ray. 2018. "*Kn0w Thy Doma1n Name*": Unbiased Phishing Detection Using Domain Name Based Features. In *SACMAT '18: The 23rd ACM Symposium on Access Control Models & Technologies (SACMAT), June 13–15, 2018, Indianapolis, IN, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3205977.3205992


*SACMAT '18, June 13–15, 2018, Indianapolis, IN, USA*

ACM ISBN 978-1-4503-5666-4/18/06…$15.00
https://doi.org/10.1145/3205977.3205992

## 1 INTRODUCTION

### 1.1 Motivation

Phishing attacks continue to be of persistent and critical concern to users, online businesses and financial institutions. A phishing website lures users into divulging their sensitive information such as passwords, pin numbers, personal information, and credit card numbers, and uses such information for financial gains. According to current estimates, the annual financial losses due to phishing attacks surpasses $3 billion. Especially, for users, a phishing attack can mean a lot more than just financial losses as the loss of sensitive personal information has long term future ramifications as well.

The major problem in detecting phishing attacks is the adaptive nature of strategies used by the phishers. Generating a phishing website has not only become trivial, but also the attackers are able to bypass most defense strategies with relative ease. For instance, the evolution of *extreme phishing* [21], a complex form of phishing that targets the identity of users shows the severity and intensity of phishing attacks. Therefore, there is a need for developing phishing detection approaches that demonstrate robustness and resiliency against the adaptive strategies being used by the phishers.

### 1.2 Problem Statement

We focus on the general problem of determining if a target website is a phishing website or not, based on the standard definitions of a phishing website from literature [1, 5]. Typically, the content of a phishing website is textually and visually similar to some legitimate website. Based on this, the problem statement we examine is, to determine the features that quantify the attacker strategies in terms of the content found in the phishing website. Such features will be used to train a machine learning model to classify between phishing and legitimate websites.

### 1.3 Limitations of Past Work

Content-based approaches [3, 4, 7, 8, 10, 11, 14, 17, 18, 20] perform in-depth analysis of content and build classifiers to detect phishing websites. These works use features extracted from the page content as well as from third-party servers, search engines and DNS servers. However, these approaches are not efficient due to the large number of training features and the dependence on third-party servers. Using third-party servers violates user privacy. Furthermore, in most these approaches, except [10], there is a critical issue of using biased datasets (see Section 1.4 for detailed discussion) and the design of features that seem to work well for such datasets.

The URL-based approaches [2, 6, 9, 16] analyze various features based on the target URL such as length of the URL, page rank of the URL, presence of special characters in the URL, host name features like IP address, DNS properties, and geographic properties. While the intuition in these approaches is sound, *i.e.*, the URL is a good indicator of phishing attacks, the structural changes of modern day

URLs negates several lexical features identified by these approaches. For instance, these days, the URLs generated by websites like Google and Amazon, are long and contain many non-alphabetic characters, which dilute the lexical similarity of legitimate URLs. For this reason, the URL based approaches inadvertently tend to be biased towards the datasets being used and are likely to be ineffective in the future. A few hybrid detection mechanisms [15, 19] combine content and URL features, but suffer from the same problems.

### 1.4 Bias in Datasets

There are two reasons for bias in datasets: *dataset usage* and *URL based features*. First, to create a labeled dataset, many researchers [8, 14, 16–18, 20] used `Alexa.com` website to create the list of legitimate websites. `Alexa.com` publishes the list of highly ranked *domain names* and a researcher generates the dataset based *only* on the index pages of these ranked websites. But, for the phishing dataset, they used anti-phishing sites like `PhishTank.com` or `Openphish.com`, which list the entire URLs of the phishing web pages. For instance, many phishers use `000webhost.com`, a free hosting service to host their phishing sites whereas this domain name itself is ranked highly in `Alexa.com`. For instance, for a feature defined as *number of sub-domains* in the URL, the legitimate URL instances obtained from `Alexa.com` will not have any sub-domains while many of phishing URL have sub-domains.

The second reason is that URL based detection [2, 6, 9, 16] does not guarantee good distinguishers between legitimate and phishing URLs. This is because adversaries have complete control over URL composition, excepting domain name, and can obfuscate against any number of measures. For instance, features like URL length, number of dots (".") in URL, presence of special characters *etc.* can easily be manipulated by phishers. In fact, this the reason for the high true negative rate (TNR) in existing works.

Except the work by Marchal *et al.* [10] where they used unbiased datasets made available by Intel security, no other work in literature has specifically addressed this concern. Furthermore, our work achieves similar classification accuracy with only seven features compared to the 200+ features used in [10].

### 1.5 Proposed Approach

Our work is the first solution to be entirely focused on the domain name of the phishing website. In our work, the domain name is the string before the top-level domain identifier, *e.g.*, for the URL `google.co.uk`, the domain name is `google`. We only concern ourselves with examining the landing page of this website, and with the information that can be extracted from this page without the help of third-party servers, search engines or DNS servers.

Our approach is based on the intuition that the domain name of the phishing websites is a key indicator of a phishing attack. We design several features that are based on the domain name and train a machine learning classifier based on sample data. The trained classifier is used to test a suspicious website against these features. Next, we describe the key challenges in our proposed approach and our solutions.

The primary challenge is to justify the design of domain name based features. Towards this end, first, we highlight the subtle distinctions between impact of the domain name and the URL of a phishing website. A phisher has much control over the formation and structure of the URL and therefore, can generate noisy URLs that can bypass most machine learning approaches. On the other hand, the phisher has limited control over the domain name, *i.e.*, the adversary can generate several types of URLs with the same domain, but the domain name remains fixed throughout. Second, domain name based features are likely to be more independent of the content in the phishing pages. The structure of the page layout, the HTML tags and the dynamic content will no longer be a major part of the detection algorithm. Third, a phishing domain name typically can contain additional characters or numbers to give the illusion of a legitimate website, *e.g.*, `gooogle.com`. These variations are subtle and are likely to provide sufficient statistical distinctions between legitimate and phishing websites. Hence, based on these arguments, we claim that the domain name based features are likely to exhibit more regularity than URL based features.

The next challenge is that the detection features could be data driven, that is, they can be biased with respect to the training data. To address this, our features mainly model the relationship between the domain name and the *visible content* of the web page. For instance, one feature calculates the rank of the domain name against all visible words on the web page, which is low for almost all phishing websites, as the attacker doesn't wish to reveal the suspicious domain name to the user. Importantly, such feature design ensures that an attacker finds it difficult to tamper with these features without arousing the suspicion of the user. But, designing such features is non-trivial and requires deep analysis of the phishing websites over a period of time. Therefore, for higher detection accuracy, we combine them with other statistical features based on observations of phishing domain names reported on `PhishTank.com` and from observations in existing research. Also, since our features are based on the domain names, we redesign the existing features to derive new features that correlate with the domain name of the website.

The penultimate challenge concerns the validity of the features. We performed a statistical validation against a small sample of the data to verify the utility of the features across the phishing and legitimate websites. We were able to eliminate several features and our final classifier consists of only seven features.

The final challenge is testing the resiliency of the domain based features to detect unknown or zero-day phishing attacks. To address this, we tested the classifier against a blacklist of URLs taken from the latest updates on `OpenPhish.com`. Our approach showed excellent resiliency and was able to detect up to 99.7% of the URLs.

### 1.6 Key Contributions

Our key contributions are: (a) We describe a machine learning (ML) based approach for phishing detection that relies entirely on domain name based features. Our approach is the first approach that has the combination of several benefits such as not using third party servers, search engines, suspicious words and URL specific features. (b) Our approach achieves 97% accuracy on a set of 2000 URLs with a five-fold cross-validation. (c) Our approach achieves 97-99.7% detection rate on live blacklist data from `OpenPhish.com`, validating our base hypothesis of bias in datasets and at the same time, demonstrating the remarkable robustness of our learning model against phisher induced noise. (d) The run-time detection speed of our approach is 4 times faster for legitimate websites and 10 times faster than the state-of-the-art work [10] in this domain. (e) We demonstrate the bias induced in the learning model by certain features, such as URL length, which raises the need for revisiting many of the existing works in literature.

## 2 RELATED WORK

Cui *et al.* [4] tried to find similarities between different attacks during a 10 month study by monitoring around 19000 websites. The study showed that 90% of phishing websites have similar HTML Document Object Model (DOM) structure and over 90% of these attacks were actually replicas or variations of other attacks in the database. Hong *et al.* [20] created a dataset to make use of the well-known term frequency inverse document frequency (TF-IDF) algorithm to find the top-5 important words in a web page and cross-checked using the Google® search engine. If the website appears in the very first list of results, then it is considered genuine.

Zhang *et al.* [18] created a framework using a Bayesian approach for content-based phishing web page detection. The model takes into account textual and visual contents to measure the similarity between the protected web page and suspicious web pages. But, this process is expensive and often results in false positives.

Ma *et al.* [9] described an approach on URL classification using statistical methods to discover the lexical and host-based properties of malicious web site URLs. They use lexical properties of URLs and registration, hosting, and geographical information of the corresponding hosts to classify malicious web pages at a larger scale. However, this approach requires a large feature set and extracts host information with the help of third-party servers.

Miyamoto *et al.* [12] provide an overview of nine different machine learning techniques and analyzed the accuracy of each classifier on the CANTINA dataset [20], reporting a maximum accuracy of 91.34% using AdaBoost. Abdelhamid *et al.* [1] experimentally compare large number of ML techniques on real phishing datasets with respect to different metrics.

Xiang *et al.* [17] proposed a layered anti-phishing solution with a rich set of features. They used machine learning techniques with 15 features, based on the HTML DOM structure, search engine capabilities, and third-party services, to detect phishing attacks. The key shortcoming of this approach is that the experiments were conducted with biased datasets as discussed in Section 1.4.

In 2015, Verma *et al.* [16] described an approach based on textual similarity and frequency distribution of text characters in URLs. For instance, they examined the character frequencies in phishing URLs and the presence of suspicious words as features. However, this approach is entirely based on URLs and is likely to be biased in the modern day context.

Jain *et al.* [8] described a machine learning based approach that extracts the features from client side only. However, their method of dataset creation is biased as discussed in Section 1.4.

Al-Janabi *et al.* [2] described a supervised machine learning classification model to detect the distribution of malicious content in online social networks (OSNs). These URLs direct users to websites that contain malicious content, phishing, and scams. Their features cannot be extracted locally and cannot guarantee the security of users outside of that network during regular browsing.

Recently, Marchal *et al.* [10, 11] propose a client-side detection approach using custom datasets from Intel security and tried to eliminate bias in datasets. However, their approach uses over 200+ features for classification, which indicates a significant time for feature extraction and classification.

There has been a rise in *extreme* phishing attacks [21] on financial institutions where the phishing website mimics the legitimate website to an alarming degree. The high level of noise in such websites is likely to defeat most content-based machine learning approaches in the past. Compared to past work, our approach relies on a nominal set of features for classification and does not examine the content of the websites in depth.

## 3 DOMAIN NAME BASED FEATURES

In Figure 1, we demonstrate some of the distinguishing domain name based features of legitimate and phishing websites.

### 3.1 Feature Engineering and Validation

As far as possible, our feature design attempts to be content-agnostic, *i.e.*, the feature design attempts to model the principles of phishing attacks and reduce the dependence of the features on specific data values. Our feature set consists of two types of features: binary, *i.e.*, the feature value is 0 or 1, and non-binary, *i.e.*, the feature is real-valued. In summary, the key principle of our feature engineering is that, all features depend on the domain name of the website and the relationships, visual and statistical, of the domain name with the content of the website. These aspects ensure that our features are not affected by biased datasets and are robust to noise.

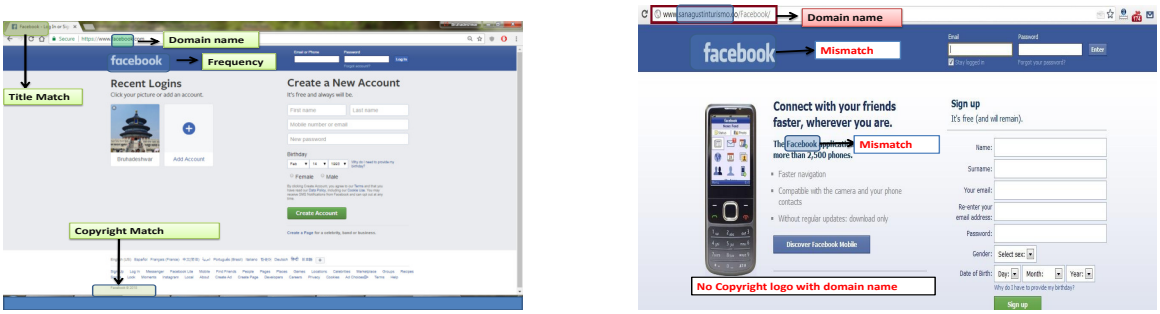
To validate the intuition behind each feature, we tested the empirical cumulative distribution function (ECDF) of the feature for 1000 phishing websites against 1000 legitimate websites. We show sample ECDF plots for a few features. We also indicate if the features are "New", meaning designed by us, or "Existing", meaning that other researchers have designed it.

### 3.2 Non-binary Features

*3.2.1 Feature 1 (New) : Domain Length.* The attackers who want to register domain for phishing have to choose longer domain name in comparison with the legitimate website. The length of domain name is the number of characters in the domain name string. As shown in Figure 2(a), the ECDF of this feature shows sufficient distinction between the legitimate websites and the phishing websites.

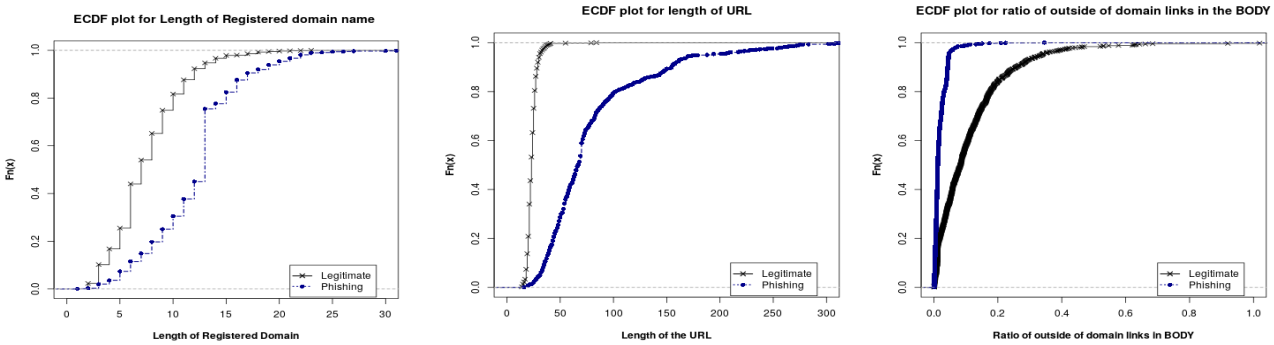
*3.2.2 Feature 2 (Existing) : URL Length.* The URL length is a popular feature among all known phishing detection approaches and is based on the intuition that phishing URLs are longer than legitimate URLs. We describe this feature here primarily to highlight the issue of dataset bias discussed in Section 1.4. In Figure 2(b), we show the ECDF of this feature. On the surface, it seems an excellent feature, however, it is completely data dependent and most existing works have generated results that are likely to be heavily influenced by the distribution of this feature in the phishing and legitimate datasets. We generated two sets of classification results: with and without the URL length, to demonstrate the impact of classification due to this feature. The average accuracy of classification increases by 2% because of this feature and reaches 99%, which matches the state-of-the-art result when only accuracy is considered. Furthermore, if the feature extraction time is also considered, we show that our results are better than the state-of-the-art.

*3.2.3 Feature 3 (Existing) : Link Ratio in BODY.* This feature is defined as the ratio of the number of hyper-links pointing to the same domain to the total number of hyper-links on the web page. The intuition is that, in the process of making a phishing website similar to the legitimate website, the attackers refer the hyper-links on the landing page to a legitimate domain name, which is different from the domain name displayed in the address bar of the browser. This feature is content-agnostic as the ratio can computed for any phishing website that exhibits this behavior. For example,




(a) Legitimate site (b) Phishing site

**Figure 1: Domain name features for legitimate and phishing websites**




(a) Domain Name Length (b) URL Length (c) Link Ratio in BODY

**Figure 2: Domain Name Length, URL Length, and Link Ratio in BODY**

the phishers create a phishing page to mimic a well-known payment service where all links on the page are to a legitimate website except the login-form in which the users need to enter their information. Accordingly, the ratio of the links referring to current domain compared to all links found in the website will be different when compared between a phishing website and a legitimate website. To evaluate this feature, we find all of links in the page and the ratio of links referring to the current page over the number of all links found on the page. However, some legitimate websites also exhibited this behavior and therefore, we used a scaling process to derive the final value of the feature. For instance, if for a given website the ratio was in the range $[0.1, 0.2]$, we assigned the value 20 to this feature. Figure 2(c), shows the ECDF of this feature, of the raw ratios, with sufficient separation between the two distributions.

*3.2.4 Feature 4 (New) : Frequency of Domain Name.* This feature counts the number of times the domain name appears as a word in the visible text of the web page. The intuition is that many web pages repeat the domain name several times in their web page, as part of disclaimers, privacy terms and so on. Therefore, if the domain name does not appear at all in the web page, then there is something suspicious about such a web page. This is a key feature that captures the visual relationship of the domain name to the web page. In practice, we find this feature to be very indicative and useful in detecting phishing websites. Note that, for classification purpose, we converted this feature into a binary feature, *i.e.*, if the domain name does not appear in the web page, we set it to 0 and if it appears more than once, we set it to 1.

**Table 1: Binary Feature Distribution**

| Feature | Legitimate | Phishing |
|---|---|---|
| HTTPS Present | 0.92 | 0.23 |
| Non-alphabetical Characters | 0.05 | 0.36 |
| Copyright Logo Match | 0.26 | 0.0 |
| Page Title Match | 0.87 | 0.03 |

## 3.3 Binary Valued Features

Table 1 summarizes the percentage distribution of the binary features in the sample dataset.

*3.3.1 Feature 5 (Existing) : HTTPS Present.* An SSL certificate is issued for a particular domain name. Most legitimate websites used SSL certificates and operated over HTTPS protocol. Therefore, if a website uses HTTPS, the feature value is 1 and if not, it is 0. Recently, phishing websites are using HTTPS as well and this explains the relative high distribution.

*3.3.2 Feature 6 (New) : Non-alphabetical Characters in Domain Name.* Attackers use non-alphabetical characters, like numbers or hyphen, to generate newer phishing domain names, which are very similar to legitimate domain names. If the domain name has any non-alphabetic character, this feature is set to 1 and 0, otherwise. Past works [8, 16] have considered a variant of this feature, *i.e.*, they examined the number of special characters in the entire URL. However, as discussed earlier, generating customized noisy URLs is a relatively easy task for the attackers.

*3.3.3 Feature 7 (New) : Domain Name with Copyright Logo.* Many legitimate websites use the copyright logo to indicate the trademark ownership on their organization name. Usually, the domain name is placed before or after the copy right logo for such websites. To generate this feature, we considered up to 50 characters before and after the copyright logo, removed the white spaces, and checked for the presence of the domain name in the resulting string. Surprisingly, we found that none of the phishing websites placed their actual domain names along with the copyright logo. To do so, would have aroused the suspicion of any web user and therefore, we found this feature to be an excellent distinguisher.

*3.3.4 Feature 8 (New) : Page Title and Domain Name Match.* Many legitimate websites repeat the domain name in the title of web page. We found that many phishing websites used this feature to deceive users into believing that they were visiting legitimate websites. But, clearly, a phishing website would not use the phishing domain name in the title page as it would be clearly visible to the user. As shown in Table 1, our intuition proved right and we found that less than 3% of the phishing websites were using this feature, but over 87% of legitimate websites had this feature.

**A Comparison with [10].** In [10], although the authors have alluded to the use of the domain name as *one* of the factors and described several features, they did not base their approach entirely on this aspect as we have done in our work. Some of the features common with our work are Feature 4, the frequency of occurrence of domain name, and Feature 8, the match of domain name with title along with some more domain name based features. Furthermore, the approach in [10] uses many other features, over 200, to perform the final classification and even ignored some domain name based features. For instance, they ignored Feature 7, domain name match with copyright logo, which we found very useful in detecting phishing websites.

# 4 EXPERIMENTAL EVALUATION

## 4.1 Experimental Methodology

We conducted two sets of experiments to assess the performance of our model trained with various machine learning classifiers. The first set of experiments were conducted on a prepared dataset and the second set of experiments were conducted on live unknown phishing dataset from `OpenPhish.com`. Only one past work [16] demonstrated a similar result on unknown datasets with a detection rate of 95%. In contrast, our approach achieves much higher detection accuracy, close to 99.7%.

During classification, total phishing websites correctly classified are denoted by, true positive (TP) and incorrectly classified as legitimate sites are denoted by, false negatives (FN), and total legitimate sites correctly classified are denoted by, true negatives (TN) and incorrectly classified as phishing websites are denoted by, false positive (FP). We report the standard classification metrics such as, positive predictive value, $PPV = \frac{TP}{TP+FP}$; true positive rate, $TPR = \frac{TP}{TP+FN}$; and accuracy, $ACC = \frac{TP+FN}{TP+FP+FN+TN}$. We show the time taken to extract the feature values for each website, the training time for each classifier, and the time taken by the classifier to predict whether a website is phishing or not.

We implemented our approach using the Sci-kit [13] library in Python 2.7 on a desktop running Fedora 24 OS with Intel Core® 2 Duo CPU E8300© 2.4 GHz processor with 6 GB RAM.

## 4.2 Datasets

For the list of legitimate websites, we obtained the 1000 top ranked websites from the `Alexa.com` and assumed them as legitimate. For the phishing websites, we got 1000 phishing websites from `PhishTank.com` and 2013 phishing websites from `OpenPhish.com`. Data was collected during the first week of January 2018.

**Dataset 1: DS-1** This set includes 1000 legitimate websites from `Alexa.com` and 1000 phishing websites from `PhishTank.com`. In the experiments, we trained and tested on this dataset with 80% data for training and 20% data for testing using five-fold cross validation.

**Dataset 2: DS-2** This dataset includes 1000 legitimate websites from `Alexa.com` and 3013 phishing websites from `PhishTank.com` and `OpenPhish.com`. For this dataset, we considered 1000 legitimate and 1000 phishing websites for training without cross-validation. The remaining 2013 websites were used for testing.

## 4.3 Experiment 1: Performance on DS-1

We designed two different experiments to evaluate the accuracy of classifiers on DS-1. In the first experiment, we used all the features described in Section 3 except URL length. In the second experiment, to show bias of URL based features, we included URL length and demonstrated the increase in classification accuracy. The URL length feature is one such biased feature that exhibits significantly different distribution for phishing and legitimate URLs, as phishing URLs are typically longer in publicly available datasets.

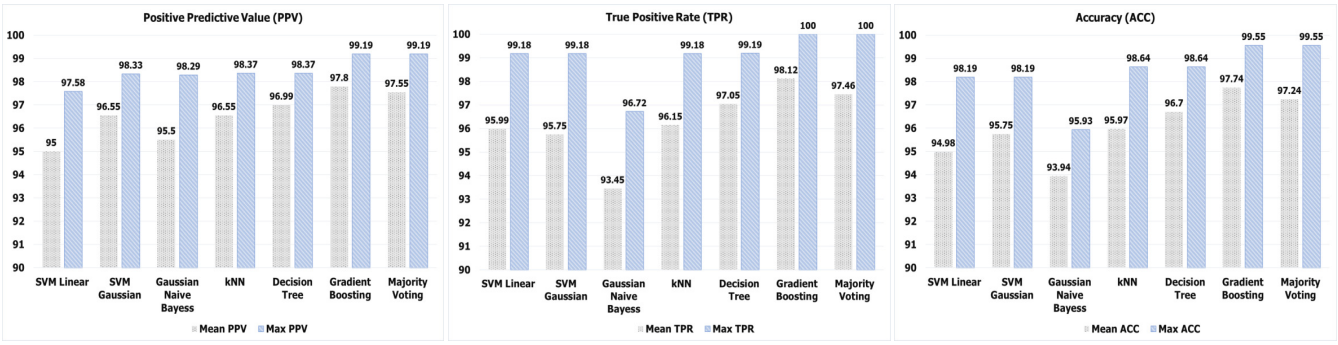
**Results without URL Length Feature.** *Our domain name based approach achieves 97% accuracy and validates our basic hypothesis.* We show the results in Figure 3. For each of the parameters, we show the maximum value achieved and the average value across all the validations. Gradient Boosting performed the best with a maximum accuracy of 99.55% percentage and an average accuracy of 97.74%. For Gradient Boosting and Majority Voting, the TPR is very high, 98.12% and 97.46%, respectively, and so is the PPV, 97.8% and 97.55%, respectively, showing the high phishing detection capability of the classifiers. We note that, our average accuracy of 97.74% is very high when compared several existing works that used a rather large and diverse set of features.

**Results with URL Length Feature.** *This feature results in higher accuracy and clearly demonstrates the bias due to the dataset.* We show the results of these experiments in Figure 4. There is an increasing trend across all the classifiers for all the parameters considered. There is clear increase in PPV where four classifiers reported an average of 98% and above with Majority Voting reporting 99%. Excepting Gaussian Naive Bayes, all other classifiers recorded an average TPR of 98% and above, with the maximum of 100% for three classifiers. The accuracy also showed an increasing trend with the average accuracy increasing to 98.8% for Gradient Boosting, and the maximum accuracy of 99.55% for several other classifiers. This experiment clearly shows that features like URL length tend to impact classification accuracy depending on the dataset.

## 4.4 Timing Analysis for DS-1

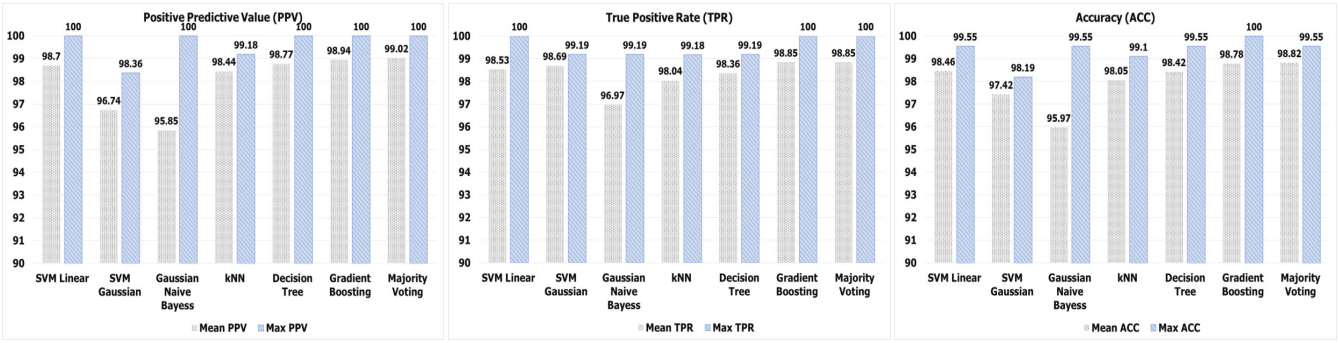
**Feature Extraction Timings.** *Our feature extraction time is very low, of the order of few milli-seconds, and demonstrates the efficiency of our feature set.*

Table 2 shows the results of our feature extraction. The total time for extracting features of a legitimate website is about 0.117 seconds and for a phishing website is 0.02 seconds, which indicates the real-time nature of our approach. This is extremely low compared to the




(a) PPV (b) TPR (c) ACC

**Figure 3: PPV, TPR and ACC on DS-1 without URL Length Feature**




(a) PPV (b) TPR (c) ACC

**Figure 4: PPV, TPR and ACC on DS-1 with URL Length Feature**

state-of-the-art approach in [10] where the extraction time was in the order of a few seconds. We emphasize that the average loading time of a web page like `msn.com`, is around 1 second and our feature extraction adds only a few milliseconds overhead to this process.

**Training and Classification Timings.** *Our classifier training and classification times are very low, of the order of few micro-seconds, and again demonstrates the efficiency of our approach.*

The testing times reported are the average across the five-fold cross validation and do not include the feature extraction time. The training can be done off-line and the testing takes a few micro-seconds to perform, after the feature extraction. Given that cumulative time for feature extraction and testing is less than 2 milliseconds, we claim that our approach can be deployed in practice as a client-side browser plug-in.

## 4.5 Experiment 2: Performance on DS-2

In this experiment, we examine robustness of our learning approach on unknown and unseen data. We obtained a list of 2013 live phishing websites from `OpenPhish.com`. Although, a higher number of sites were listed, many sites were unavailable and few were blocked by the corresponding ISPs. We trained the classifier in two modes: without including the URL length feature and with the URL length feature included. Finally, we tested the resulting classifier on the 2013 data instances and show the results in Table 4. These results show the remarkable performance of our approach. Unlike the previous approach [16], which attempted a similar experiment, for many of our classifiers, the TPR largely remains *unchanged* across both the experiments and even shows a slight increase for Decision tree and Gradient Boosting classifiers. Furthermore, when including URL length, the TPR even reaches 99.7%(!) for *k*NN and Gradient Boosting. This result also confirms our hypothesis that domain name based features can accurately capture the nature of a phishing website.

## 4.6 Comparison with Previous Work

We compare our results empirically with existing state-of-the-art solutions in Table 5. Our basis for comparison is the number of features, the accuracy, whether client-side features only are used or third-party features are included and average accuracy. We did not include the run-times of the approaches as that is a system specific metric. However, we note that our scheme reports micro-second level feature extraction and classification time, even when run on a relatively low performance laptop with Core 2 Duo processor.

# 5 CONCLUSION

In this work, we described the first approach towards the design of *only* domain name based features for detection of phishing websites using machine learning. Our feature design emphasized on the elimination of the possible bias in classification due to differently chosen datasets of phishing and legitimate pages. Our approach differs from all previous works in this space as it models the relationship of the domain name to the intent of phishing. With only seven features we are able to achieve a classification rate of 97% with cross-validated data. Furthermore, we were able to show a detection rate of 97-99.7% for live black-listed URLs from `OpenPhish.com`.

**Table 2: Feature Extraction Timings**

| Feature | Legitimate ($\mu$s) | Phishing ($\mu$s) |
|---|---|---|
| HTTPS Present | 4.12 | 3.87 |
| Domain Length | 63.45 | 66.45 |
| Page Title Match | 26.9 | 32.3 |
| Frequency Domain Name | 333.8 | 33.09 |
| Non-alphabetic Characters | 32.64 | 13.68 |
| Copyright Logo Match | 2737.56 | 450.48 |
| Link Ratio in Body | 114482.87 | 19445.67 |
| URL Length | 0.3576 | 0.5066 |
| Total Time (in seconds) | 0.117 | 0.02 |

**Table 3: Training/Testing Timings**

| Classifier | Train (in ms) | Test (in $\mu$s) |
|---|---|---|
| SVM Linear | 1339.85 | 6.74 |
| SVM Gaussian | 703.62 | 38.32 |
| Gaussian Naive Bayes | 2.28 | 1.47 |
| kNN | 7.36 | 14.85 |
| Decision tree | 2.49 | 0.80 |
| Gradient Boosting | 2737.56 | 450.48 |
| Majority Voting | 177.73 | 3.25 |

**Table 4: True Positive Rate for DS-2**

| Classifier | Without URL Length | With URL Length |
|---|---|---|
| SVM Linear | 94.09 | 94.24 |
| SVM Gaussian | 92.75 | 90.81 |
| Gaussian Naive Bayes | 91.06 | 92.75 |
| kNN | 93.74 | **99.7** |
| Decision tree | 97.91 | 97.27 |
| Gradient Boosting | **98.21** | **99.75** |
| Majority Voting | 95.33 | 97.67 |

**Table 5: Comparison with State-of-the-art Approaches**

| Approach | ♯ of Legitimate sites | ♯ of Phishing Sites | ♯ of features | Accuracy | Client Side |
|---|---|---|---|---|---|
| Cantina [20] | 2100 | 19 | 7 | 96.97 | No |
| Cantina+ [17] | 1868 | 940 | 15 | 97 | No |
| Verma *et al.* [16] | 13274 | 11271 | 35 | 99.3 | Partial |
| Off-the-Hook [10] | 20000 | 2000 | 210 | 99.9 | Yes |
| Our approach without URL Length | 1000 | 3013 | 7 | 97.7 | Yes |
| Our approach with URL Length | 1000 | 3013 | 8 | 98.8 | Yes |

This shows that our approach is able to adapt to the complex strategies used by phishers to evade such detection mechanisms. As our features explore the content found in the visible space of the web page, an attacker will need to put a huge effort to bypass our classification. In trying to bypass our approach, an adversary may end up designing a page that will make any user suspicious. Furthermore, we demonstrated the shortcoming of using URL features such as URL lengths, that seem to give higher accuracy but may not do so in the near future. Our feature extraction and classification times are very low and show that our approach is suitable for real-time deployment. In future, we wish to explore the robustness of machine learning algorithms for phishing detection in the presence of newer phishing attacks. We are also developing a real-time browser add-on that will provide warnings when visiting suspicious sites.

## ACKNOWLEDGEMENT

This work was supported in part by funds from NSF under Award No. CNS 1650573, CableLabs, AFRL, Furuno Electric Company, and SecureNok.

## REFERENCES

[1] Neda Abdelhamid, Fadi A. Thabtah, and Hussein Abdel-jaber. 2017. Phishing Detection: A Recent Intelligent Machine Learning Comparison Based on Models Content and Features. In *Proc. of the IEEE Int. Conf. on Intelligence and Security Informatics (ISI)*. 72–77.

[2] Mohammed Al-Janabi, Ed de Quincey, and Peter Andras. 2017. Using Supervised Machine Learning Algorithms to Detect Suspicious URLs in Online Social Networks. In *Proc. of the IEEE/ACM Int. Conf. on Advances in Social Network Analysis and Mining (ASONAM)*. 1104–1111.

[3] Ram B. Basnet, Srinivas Mukkamala, and Andrew H. Sung. 2008. Detection of Phishing Attacks: A Machine Learning Approach. In *Soft Computing Applications in Industry. Studies in Fuzziness and Soft Computing*. Vol. 226. Springer, 373–383.

[4] Qian Cui, Guy-Vincent Jourdan, Gregor V Bochmann, Russell Couturier, and Iosif-Viorel Onut. 2017. Tracking Phishing Attacks over Time. In *Proc. of the Int. World Wide Web (WWW) Conf.* 667–676.

[5] Z. Dou, I. Khalil, A. Khreishah, A. Al-Fuqaha, and M. Guizani. 2017. Systematization of Knowledge (SoK): A Systematic Review of Software-Based Web Phishing Detection. *IEEE Communications Surveys Tutorials* 19, 4 (2017), 2797–2819.

[6] Sujata Garera, Niels Provos, Monica Chew, and Aviel D Rubin. 2007. A Framework for Detection and Measurement of Phishing Attacks. In *Proc. of the ACM Workshop on Recurring Malcode (WORM)*. ACM, 1–8.

[7] R. Gowtham and Ilango Krishnamurthi. 2014. A Comprehensive and Efficacious Architecture for Detecting Phishing Webpages. *Computers and Security* 40 (2014), 23–37.

[8] Ankit Kumar Jain and B. B. Gupta. 2017. Towards Detection of Phishing Websites on Client-side Using Machine Learning Based Approach. *Telecommunication Systems* (December 2017), 1–14.

[9] Justin Ma, Lawrence K. Saul, Stefan Savage, and Geoffrey M. Voelker. 2009. Beyond Blacklists: Learning to Detect Malicious Web Sites from Suspicious URLs. In *Proc. of the ACM Int. Conf. on Knowledge Discovery and Data Mining (KDD)*. ACM, 1245–1254.

[10] Samuel Marchal, Giovanni Armano, Tommi Grondahl, Kalle Saari, Nidhi Singh, and N. Asokan. 2017. Off-the-Hook: An Efficient and Usable Client-Side Phishing Prevention Application. *IEEE Trans. on Computers* 66, 10 (2017), 1717–1733.

[11] Samuel Marchal, Kalle Saari, Nidhi Singh, and N Asokan. 2016. Know Your Phish: Novel Techniques for Detecting Phishing Sites and Their Targets. In *Proc. of IEEE Int. Conf. Distributed Computing Systems (ICDCS)*. IEEE, 323–333.

[12] Daisuke Miyamoto, Hiroaki Hazeyama, and Youki Kadobayashi. 2008. An Evaluation of Machine Learning-based Methods for Detection of Phishing Sites. In *Proc. of the Int. Conf. on Neural Information Processing (ICONIP)*. Springer, 539–546.

[13] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, and others. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12, Oct (2011), 2825–2830.

[14] Routhu Srinivasa Rao and Alwyn Roshan Pais. 2018. Detection of Phishing Websites using an Efficient Feature-based Machine Learning Framework. *Neural Computing and Applications* (January 2018).

[15] Choon Lin Tan, Kang Leng Chiew, KokSheik Wong, and San Nah Sze. 2016. PhishWHO: Phishing webpage detection via identity keywords extraction and target domain name finder. *Decision Support Systems* 88, C (2016), 18–27.

[16] Rakesh Verma and Keith Dyer. 2015. On the Character of Phishing URLs: Accurate and Robust Statistical Learning Classifiers. In *Proc. of ACM Conf. on Data and Applications Security and Privacy (CODASPY)*. 111–122.

[17] Guang Xiang, Jason Hong, Carolyn P. Rose, and Lorrie Cranor. 2011. CANTINA+: A Feature-Rich Machine Learning Framework for Detecting Phishing Web Sites. *ACM Trans. Information and Systems Security (TISSEC)* 14, 2 (September 2011), 1–28.

[18] Haijun Zhang, Gang Liu, Tommy W. S. Chow, and Wenyin Liu. 2011. Textual and Visual Content-Based Anti-Phishing: A Bayesian Approach. *IEEE Trans. on Neural Networks* 22, 10 (2011), 1532–1546.

[19] Wei Zhang, Qingshan Jiang, Lifei Chen, and Chengming Li. 2017. Two-stage ELM for Phishing Web Pages Detection Using Hybrid Features. *World Wide Web* 20, 4 (2017), 797–813.

[20] Yue Zhang, Jason I Hong, and Lorrie F Cranor. 2007. Cantina: A Content-based Approach to Detecting Phishing Web Sites. In *Proc. of the World Wide Web (WWW) Conf.* ACM, 639–648.

[21] Rui Zhao, Samantha John, Stacy Karas, Cara Bussell, Jennifer Roberts, Daniel Six, Brandon Gavett, and Chuan Yue. 2017. Design and Evaluation of the Highly Insidious Extreme Phishing Attacks. *Computers & Security* 70 (2017), 634 – 647.