URL-based Phishing Detection using the Entropy of Non-Alphanumeric Characters

Eint Sandi Aung
Department of Computer Science and Communications Engineering
Waseda University
Tokyo, Japan
eintsandiaung@toki.waseda.jp

Hayato Yamana
Department of Computer Science and Communications Engineering
Waseda University
Tokyo, Japan
yamana@waseda.jp

ABSTRACT

Phishing is a type of personal information theft in which phishers lure users to steal sensitive information. Phishing detection mechanisms using various techniques have been developed. Our hypothesis is that phishers create fake websites with as little information as possible in a webpage, which makes it difficult for content- and visual similarity-based detections by analyzing the webpage content. To overcome this, we focus on the use of Uniform Resource Locators (URLs) to detect phishing. Since previous work extracts specific special-character features, we assume that non-alphanumeric (NAN) character distributions highly impact the performance of URL-based detection. We hence propose a new feature called the entropy of NAN characters for URL-based phishing detection. Experimental evaluation with balanced and imbalanced datasets shows 96% ROC AUC on the balanced dataset and 89% ROC AUC on the imbalanced dataset, which increases the ROC AUC as 5 to 6% from without adopting our proposed feature.

CCS CONCEPTS

• Security and privacy→Phishing • Information systems→Content analysis and feature selection

KEYWORDS

Phishing, Webpage, URL, Detection

ACM Reference format:

EintSandi Aung and Hayato Yamana. 2019. URL-based Phishing Detection using the Entropy of Non-Alphanumeric Characters. In *Proceedings of the 21st International Conference on Information Integration and Web-based Applications & Services (iiWAS2019). Munich, Germany, 8 pages.* https://doi.org/10.1145/3366030.3366064

iiWAS2019, December 2-4, 2019, Munich, Germany

ACM ISBN 978-1-4503-7179-7/19/12…$15.00
https://doi.org/10.1145/3366030.3366064

1 Introduction

Cyber phishing is no longer an unfamiliar topic since the Internet has become an essential tool used on a daily basis. People can do almost anything at home while sitting in front of their personal computer – from grocery shopping to accessing health support services. It has been reported that the number of the Internet users has significantly increased from nearly 0.15 billion in 1998 to 4.32 billion in 2018, which is a growth of approximately 30 times [1]. This indicates that people strongly depend on the Internet on a daily basis. At the same time, this enormous amount of information has tempted some people to sneak onto the Internet and commit phishing crimes. The report [1] shows that phishers commit crimes not only for money, but for fame, acknowledgement, or just out of curiosity. For these reasons, phishing detection has drawn attention from researchers.

Phishing is a cyber threat in which attackers take advantage of users by mimicking legitimate authentic websites in order to steal sensitive information such as passwords and bank information. Phishing is performed through different media such as the Internet, short message service, and voice. Their target vectors can be email, instant messaging, smishing (short message phishing), vishing (voice phishing), or websites [2]. In this paper, phishing refers to web phishing through the Internet. Although phishing can be defended against by both 1) user awareness and 2) technology-based approaches, the former approach cannot be completely trusted because it relies on humans – not all of whom are aware of phishing.

The Anti-Phishing Working Group [2] stated that the total number of unique phishing websites detected was approximately 785,000 in 2018, which is a significant number. They also said that the use of web page redirects to make actual phishing sites look like legitimate sites to victims has increased. When potential victims click on these links, they are redirected to phishing sites via several

[1] Internet Growth Statistics by Internet World Stats. Today's road to e-commerce and global trade internet technology reports.

[2] Anti-phishing Working Group APWG. Unifying the global response to cybercrime. Phishing activity trends report, 4th Quarter 2018.

intermediate sites. Then, phishers request credential information or install malware on the victim's computer.

Moreover, according to PhishLabs [3] , HTTPS phishing sites have become popular in the past two years. At the end of 2016, less than 5% of the HTTPS infrastructure consisted of phishing sites. By Q4 of 2017, it had increased to 33%. Phishers target users who have a poor awareness of the green lock (Figure 1) and act as if the website is secured.

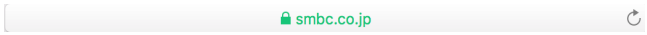

Figure 1: Green lock icon

In fact, the green lock only shows that the website uses Secure Sockets Layer (SSL) certificates; it does not guarantee the security of the site. In addition, phishers mimic websites so that they are similar to legitimate ones, without providing much content information. This can be analyzed by researchers for detection purposes. The phishers hide suspicious information as much as possible to lure users. We can summarize the problem statement as follows:

1. Users are tricked by Uniform Resource Locators (URLs) because of their lack of knowledge.
2. Phishers reveal the information of a webpage's content as little as possible.

We assume that non-alphanumeric (NAN) characters are useful for phishing detection because phishers tend to create fake URLs with NAN characters such as:

1. extra unnecessary dots
2. "//" to redirect the user to a completely different domain
3. "-" in the domain to mimic a similar website name
4. unnecessary symbols

However, instead of directly using the frequencies of NAN characters found in URLs, we propose a new feature to measure the distributions of these special characters between phishing and legitimate websites. We propose a feature called the *entropy* of NAN characters for URL-based phishing detection. Our objective is to develop a new feature that is useful in URL-based phishing detection whenever little or no information is available in a webpage of the phishing website. This is because some of the attackers hide information on the page's content, which is difficult to detect by other approaches such as content-based detection. The attackers do not show any valuable information, but a mere visually similar design, which makes users difficult to differentiate with the legitimate websites. To overcome this, we propose the new entropy of NAN feature to detect phishing by using URL only, without accessing the content of the webpage, as we know the participation of NAN characters in the URL greatly effects on the classification of phishing and legitimate websites.

The rest of the paper is organized as follows. Section 2 briefly explains related work for phishing detection. Section 3 details our proposed features, feature selection of NAN characters and lists of previously proposed features for phishing detection. Section 4 presents datasets and experimental evaluation. Eventually, we conclude our work with discussion for future work in Section 5.

[3] Phishlabs. a member of Anti Phishing Working Group (APWG)

2 Related Work

Although there are various media and vectors that can be categorized for phishing – the Internet as a medium, email or instant messaging as vectors, and social engineering as technical approach [2] – we can mainly classify phishing attacks into two types according to a social engineering perspective [1] as email spoofing and fake websites as shown in Figure 2.

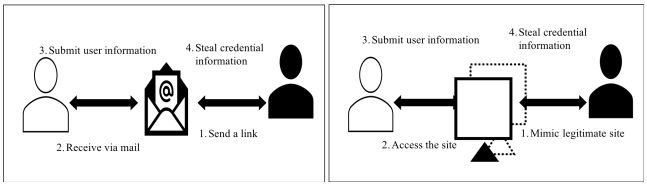

Figure 2: Email spoofing (left) and fake website (right)

We survey phishing detection approaches to better understand how they work. Phishing detection schemes are categorized differently in various papers [3,4,5,6]. Thus, we use a general categorization and list phishing detection approaches and their works in Table 1.

Table 1. Phishing detection approaches

Approach	Description
Whitelist (e.g. Automated Individual WhiteList (AIWL)[7])	Solution: 1. Maintain own individual whitelist. 2. Alerts user of a possible attack if a credential to a Login User Interface (LUI) not on the whitelist is submitted. Advantages: 1. Performs well at detection (LUI authentication) by storing all the LUI information rather than only a URL. 2. Efficiently defends against pharming attacks by alerting users when a legitimate IP address has been maliciously changed (these attacks cannot be detected by SpoofGuard [8]). Limitations/Remarks: 1. The system warns whenever any information is sent to other trusted pages that are not in the whitelist because AWIL maintains a list of previous successful LUIs. 2. Difficult to defend against Trojan Horse and viruses on a local machine because the whole AWIL is installed on the local PC.
Blacklist (e.g. Google Safe Browsing [9])	Solution: 1. Block a URL if it is in a blacklist.

Approach	Description
Blacklist (e.g. Google Safe Browsing [9])	Advantages: 1. Privacy: hashed URLs are used to preserve the actual URL queried by user. Limitations/Remarks: 1. Zero-day attacks and IP changes cannot be detected.
Content/ Heuristic (e.g. PhishGuard [10])	Solution: 1. Identify phishing websites by submitting random credentials in a login process before submitting actual credentials. Advantages: 1. Phishing websites are detected if two situations occur: 1) the number of failure responses are less than the number of attempts or 2) the number of success responses are more than 1. This is based on the fact that phishing sites only store credentials for future use and do not verify them. Limitations/Remarks: 1. Credential theft occurs if the access is unauthorized when the correct credentials have been submitted after several "http 401" responses. That is, an "http 401" error has two meanings: 1) a wrong password was submitted or 2) the website alerts authentication failure by default. 2. Alert authentication failure with "HTTP 200 OK" response by redirecting to a new page in some legitimate websites (such as ICICI bank).
Visual Similarity (e.g. [11])	Solution: 1. Detect phishing by finding similarities between phishing and legitimate websites based on text attributes, hidden images, and overall visual appearance. Advantages: 1. It integrates its visual similarity detection techniques with the open source tool AntiPhish [12] to overcome the problem that it flags legitimate credential re-use as suspicious. Limitations/Remarks: 1. This approach cannot distinguish if text is replaced with an image with the same appearance.
URL analysis (e.g. [13])	Solution: 1. Use statistical (e.g., mean and median between URL features) and lexical (e.g., title and text content) features to identify phishing websites. Advantages: 1. Webpages in different languages can be detected. 2. Zero-hour phishing attacks can be detected. Limitations/Remarks: 1. The Alexa rank feature gives phishing domains (e.g., 000webhost.com) a high rank.

We adopt URL-based phishing detection out of all other approaches because our objective is to detect phishing without accessing the webpage. Moreover, only URL-based approach can perform zero-hour detection. Thus, we describe URL-based detection compared with other approaches as follows in Table 2.

Table 2. Comparison of phishing detection approaches

Approach	Access to URL (downloading the content)	Zero-hour detection
Whitelist	Yes	No
Blacklist	Yes	No
Content/Heuristic	Yes	No
Visual Similarity	Yes	No
URL	No	Yes

3 URL-based Phishing Detection

To enhance the previously proposed URL-based phishing detection schemes [13,14], we propose a new feature called the *entropy* of NAN characters. Inspired by previous studies [13, 14] that adopt the frequencies of specific special characters (called NAN characters here) such as "-", "//", "_", and "." in each URL as features, we have a hypothesis that the number of times phishing websites use these special characters in a URL has different statistics in comparison with legitimate websites.

An overview of our URL-based phishing detection system is shown in Figure 3. As shown in Figure 3, we adopt 12 features in total, of which 10 are previously proposed features from various papers (called "existing features" in this paper) and the remaining two (F11 and F12) are our proposed features, called "proposed features." The reason why we choose these 10 previously proposed features is that they do not require any contents of web page so that we do not have to access and retrieve the webpage.

3.1 Proposed Feature – Entropy of NAN Characters

The key idea of computing the entropy using NAN characters is to determine how the NAN characters are distributed in each URL. Since our hypothesis is that the level of disorder of NAN characters between phishing and legitimate websites are different, we measure

NAN character distribution using entropy – frequentist probability distribution.

3.2 Entropy of NAN Characters

Previous studies [13, 14] have extracted the frequencies of specific special characters such as "-", "//", "_", and "." in each URL. Instead of measuring the frequencies of specific special characters and representing them as distinct features, we compute their overall entropy to measure the frequentist probability distribution between phishing and legitimate websites to represent them as a single feature.

For example: instead of measuring the number of "." as feature *Fx*, the number of "-" as feature *Fy*, the number of "@" as feature *Fz*,

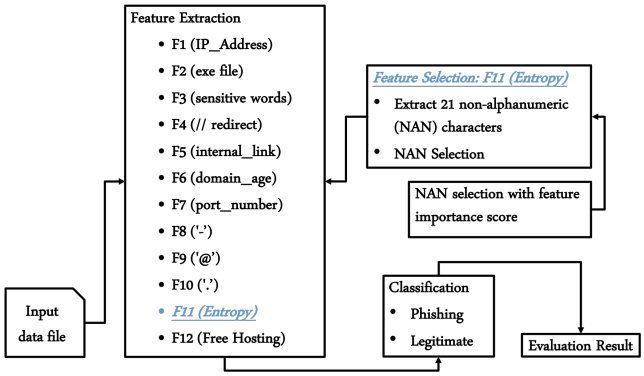

Figure 3: Our URL-based phishing detection system

and using a vector [*Fx,Fy,Fz*] as the final feature for a URL, we compute the entropy of all NAN characters as feature *Fe* and use the vector [*Fe*], where *Fe = entropy(frequentist probability distribution of NAN characters).* Since the presence of special characters in phishing and legitimate websites differs, we assume that measuring their distribution becomes a more discriminative feature for better classification. We adopt entropy because it is a measure of disorder (as well as a measure of purity). A high level of disorder means a low level of purity – if the probabilities between two special characters are not substantially different (i.e., 4/7 and 3/7), this indicates a high level of entropy or low level of purity. Otherwise, it indicates a low entropy or high purity.

We define the entropy of NAN characters as follows:

$$Entropy = -\sum_{i \in T}^{n} (P_i log P_i) \quad (1)$$

,where i is the i-th NAN character in T, i.e., $T(i)$, T is the list of NAN characters, and P_i is the frequentist probability of the i-th NAN character. Note that we use logarithmic base two in the calculation.

A high level of entropy indicates a high level of disorder. For example, assume we have five different NAN characters {"-", ".", "_", "@", "&"} whose appearance probabilities are all 0.2. Then, we have $-\sum_{i=1}^{5}(0.2 log 0.2) = 2.32$. Note that, a higher level of entropy indicates a lower level of purity. We consider that NAN characters play a huge impact on the classification of phishing websites. Thus, we use entropy to determine how each NAN character is distributed in each URL. Note that we consider the probability distribution of NAN characters at URL level, not their distribution over entire datasets, because the entropy of NAN characters becomes one of the features of the target URL to be detected as phishing or legitimate.

3.3 Selection of NAN Characters

Selection of NAN characters is performed to select important NAN characters to classify a given URL into phishing or legitimate. We choose 21 characters as the candidate NAN characters as shown in Table 3. They are defined by RFC3986 as the reserved characters that can be used in Uniform Resource Identifier (URI). We also add '%' because it can be used for percent encoding. By adopting the random forests (RF) to calculate a feature importance score with training datasets that are 70% of the whole dataset shown in Table 5, we select the top-10 NAN characters from Table 3. Since we do not know what kind of NAN characters phishers typically use, we choose effective ones before employing the entropy.

Table 3. Candidate NAN characters

Name	Symbol	Name	Symbol
hash	#	percent	%
dash	-	underscore	_
dollar	$	question	?
asterisk	*	comma	,
left parenthesis	(,[,{	equal	=
right parenthesis	),],}	ampersand	&
semicolon	;	tilde	~
colon	:	period	.
apostrophe	‘	plus	+
slash	/	at	@
exclamation	!		

The feature importance scores are shown in Figure 4. The selected top-10 NAN characters are shown in Table 4.

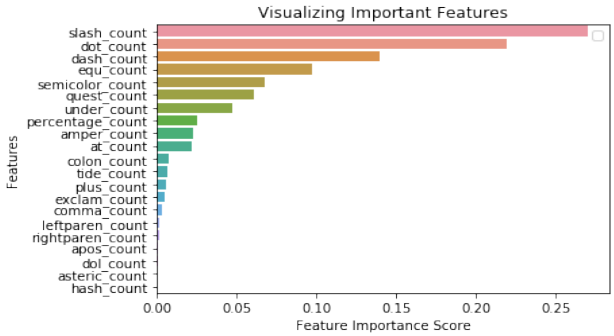

Figure 4: Feature importance scores

Table 4. Top-10 Selected NAN characters

Name	Symbol	Name	Symbol
slash	/	question mark	?
dot	.	underscore	_
dash	-	percentage	%
equal	=	ampersand	&
semicolon	;	at	@

3.4 Previously Proposed Features

Other previously proposed features, i.e., F1 to F10, adopted in our system are described below.

F1 – IP address (binary): This feature indicates whether the URL contains an IP address. It is −1 if an IP address is found; otherwise, it is 1. Our hypothesis is that phishers tend to use an IP address instead of a domain name to direct users to a phishing page to confuse users because it hides the domain to which the current link with IP address belongs (for example, http://67.205.147.248/oft/index.php?produto=722415036).

F2 – exe (binary): This feature indicates whether the URL contains an exe file. It is −1 if one is found; otherwise, it is 1. The hypothesis is that phishers tend to use exe files to run malware in the background processes.

F3 – sensitive word (binary): This feature indicates whether the URL contains sensitive words [15] such as "confirm," "account," "banking," "secure," "login," and "signin." It is −1 if these words are found, otherwise, it is 1. The hypothesis is that most of the phishers tend to use those keywords to lure victims as if the site is legitimate.

F4 - // redirect (binary): This feature indicates whether the path of the URL contains a "//" to redirect users to a phishing page. It is −1 if these symbols are found, otherwise, it is 1. The hypothesis is that "//" is mainly used for redirection by ignoring the left side of URL. (for example, http://redirect.company.com/http://exter nalsite.com/page).

F5 – internal link (binary): This feature indicates whether the path of the URL contains another link. It is −1 if one is found, otherwise, it is 1. The hypothesis is that phishers tend to add internal URLs in the main URL (for example, http://www.linkebuy.com.br/linkebuy/parceiro?protocol=http&url=www.google.com).

F6 – age of domain (binary): This feature indicates whether the age of the domain is less than 12 months. We check the "whois" property of the domain and obtain its creation date if we can successfully access it. It is −1 if the creation date is less than 12 months from the access date; otherwise, it is 1. The hypothesis is that phishing links do not exist for a long time and phishers mostly use newly created links.

F7 – port number (nonbinary): This feature indicates which port number is used. If it is 443 (HTTPS), then the feature is 1. If it is 80 (HTTP), then the feature is 0; otherwise it is −1. The hypothesis is that previous phishers have rarely used HTTPS. However, in the 2017 PhishLab report [16], it was stated that they have started using HTTPS. Thus, we differentiate HTTPS from other possibilities.

F8 – dash-symbol count (nonbinary): This feature is the number of "-" symbols in a URL. The hypothesis is that phishers tend to mimic legitimate websites by adding "-" characters in a URL.

F9 – at-symbol count (nonbinary): This feature is the number of "@" signs in a URL. The reason is that when we analyzed our phishing datasets, we found that phishers more often use "@" in a URL, especially in a URL query value, than legitimate websites (for example, http://shunmas.com/sj/index.php?email=2 @2.com).

F10 – dot-symbol count (nonbinary): This feature is the number of "." symbols in the URL. The hypothesis is that phishers either use more subdomains than legitimate websites, or unnecessary "." symbols in the URL path (for example, https://www.ssproduction .com.pk/spages/verify.php?ga=2.38170595.17086121.1551095253-653443608.1551095253&mail=laurent@tacer.biz..).

3.5 Another Proposed Feature – Free Host

Besides the entropy of NAN characters (F11), we also propose a new feature called *free host feature* shown below.

F12 – free host (binary): This feature indicates whether the URL uses a free hosting domain. The hypothesis is that phishers mainly use free hosting URLs (for example, 000webhost.com is often found in our phishing datasets). We manually surveyed the free hosting domains present in our phishing datasets and found several phishing domains using 000webhostapp.com. Any user can create a website with this domain name after signing up at its free hosting service. We also included the free hosting services that are used by phishers, as surveyed by [17] in 2012. Although [17] is no longer updated, 000webhost is still popular among phishers. Hosting services such as 000webhost, rank high in Alexa.com, however, they are mostly used by phishers. Thus, we add this new feature. The feature is −1 if the URL is managed by free host; otherwise, it is 1.

4 Experimental Evaluation

In this section, we evaluate the effectiveness of our proposed features by adopting Random Forest classifier.

4.1 Datasets

During the period April 6–8, 2019, we gathered both phishing URLs from PhishTank [18] and legitimate URLs from DMOZ (or called as Curl) – the largest human-edited directory of the Web [19]. We prepare two datasets called D1 and D2. The D1 is balanced dataset in which the same number of legitimate and phishing URLs are included. The D2 is imbalanced dataset where the ratio of legitimate URLs to phishing URLs is nearly 9:1. To ensure the URLs in the datasets present currently, we chose only active legitimate website links, by validating each URL by obtaining an HTTP response from it. As for the phishing website links, we used the latest updated URLs as much as possible. We list dataset details in the following Table 5.

Table 6. Comparison of results

Dataset	Feature	Precision Rate (Prec)	Recall Rate (R)	True Positive Rate (TPR)	False Positive Rate (FPR)	True Negative Rate (TNR)	False Negative Rate (FNR)	Accuracy (ACC)	ROC AUC	F1
D1	w/ entropy	0.90	0.90	0.90	0.10	0.90	0.10	89.82	96.20	89.82
	w/o entropy	0.83	0.80	0.81	0.19	0.81	0.19	80.68	87.51	80.28
D2	w/ entropy	0.94	0.94	0.74	0.26	0.74	0.26	94.05	89.31	93.20
	w/o entropy	0.93	0.93	0.66	0.35	0.66	0.35	92.94	84.31	91.36

Table 5. Dataset details

Dataset	Size		Type
	Legitimate	Phishing	
D1	5,000	5,000	Balanced
D2	95,754	10,473	Imbalanced

4.2 Preparation and Results

In this evaluation, we adopt Random Forest (RF) classifier to execute 10-fold cross validation. As for the hyperparameter tuning, we varied the number of trees up to 200 from 10 in increments of 10 (here, we list only from 100 to 200 because the best performance has more than 100 number of trees), and from 200 to 700 in increments of 100. For the Gini impurity, the "gini" criterion is used. For information gain, "entropy" is used. We divide the test dataset into 10 splits (split0–split9) to measure the results.

We measured true positive rate (TPR), false positive rate (FPR), precision rate (Prec), recall rate (R), F1, ROC AUC (area under the receiver operating characteristic curve), and accuracy (ACC) as shown in Table 6.

We obtain ROC AUC score 89.31% w/ entropy and 84.31% w/o entropy on the imbalanced dataset; ROC AUC score 96.20% w/ entropy and 87.51% w/o entropy. Here, w/ entropy feature set is as same as the feature set shown in Figure 3 and w/o entropy feature set omits F11(entropy) from the feature set shown in Figure 3.

4.3 Comparison of three classifiers

In this evaluation, we compare evaluation results from section 4.2 with results of two classifiers such as Gaussian Naïve Bayes (GNB) and Support Vector Machine (SVM). As GNB has no parameter to tune, we perform cross validation with 10 splits. For SVM, we perform tuning with three parameters; "C", "kernel" and "gamma". We tune four pairs of parameters; C=[0.1,1,10], gamma=['auto', 'scale'] for kernel=[linear, poly, sigmoid] and C=[0.1,1,10], gamma=['0.1, '1'] for kernel=[rbf]. We illustrate evaluation results comparing with RF in Figure 5. RF outperforms with 96.2% over 90.66% in GNB and 93.37% in SVM on balanced dataset D1 and 89.31% over 80.05% in GNB and 82.33% in SVM on imbalanced dataset D2. Furthermore, parameter tuning with SVM takes incredibly longer time (10 times longer) than RF.

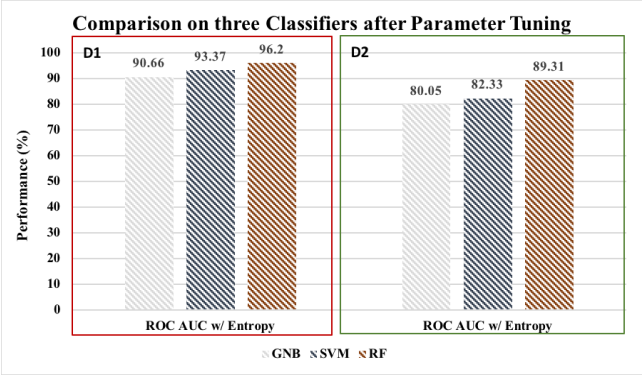

Figure 5: Comparison on three classifiers

4.4 Comparison between Entropy and Frequencies of NAN Characters

Although we proposed a new feature called the entropy of NAN characters and confirmed its effectiveness for phishing detection in Section 4.2, we will compare the entropy of NAN characters with a set of frequencies of NAN characters in a URL using Random Forest (RF) classifier. Instead of adopting the entropy, we may adopt a naïve technique, i.e., the frequency of each special character in a URL. Thus, the comparison of the entropy and the set of frequencies become important to show the effectiveness of the entropy. As shown in Table 7, we prepare two feature sets, i.e., Feature set 1 and Feature set 2. Feature set 1 consists of all the frequencies of NAN characters shown in Table 3. Note that the frequency is represented by the percentage of its appearance in a URL. Feature set 2 is same as our proposed feature set. We list two features sets in Table 7. In this experiment, we also applied the same hyperparameter tuning mentioned in Section 4.2. The

execution time was gathered on Intel (R) Core (TM) i7-4790 CPU @ 3.60 GHz with 16GiB memory.
The results in Table 8 shows that 1) ROC AUC is almost the same between feature set 1 and feature set 2, however 2) our proposed feature set 2 has faster execution time than feature set 1, because of small number of features. Thus, our feature set 2 has advance in the execution time as shown in Table 8.

Table 7. Feature sets

Feature Set Name	Features
Feature Set 1 (# of Features:34)	IP address, exe, sensitive words, "//" redirect, internal link, age of domain, port number, free hosting, percentage of ("#" count, "@" count, "-" count, "." count, "$" count, "*" count, "[" count, "(" count, "{" count, "]" count, "}" count, ")" count, "+" count, ";" count, "~" count, ":" count, "'" count, "/" count, "%" count, "?" count, "," count, "=" count, "&" count, "!" count, " " count), and total NAN character count.
Feature Set 2 (# of Features:12)	IP address, exe, sensitive words, "//" redirect, internal link, age of domain, port number, "-" count, "@" count, "." count, NAN character entropy, and free hosting.

Table 8. Comparison of feature sets 1 and 2

Dataset	Feature Set Name	ROC AUC	Execution Time (s)	Execution Time Difference between D1 and D2 (s)
D1 Balanced	Feature set 1	96.20	381.00	115.00
	Feature set 2	96.20	266.00	
D2 Imbalanced	Feature set 1	90.55	4,870.00	1,293.00
	Feature set 2	89.31	3,577.00	

5 Discussion

In this paper, we investigated phishing website detection from two perspectives: 1) a dataset perspective and 2) a feature perspective.
From the dataset perspective, we performed the detection on a balanced dataset and an imbalanced dataset, in which the majority of data consists of legitimate sites. Our assumption is that the real world dataset consists mostly of legitimate websites, so we cannot expect the same or a similar number of phishing websites with legitimate websites. Thus, we want to know if our system works well on imbalanced data.
From the feature perspective, we contributed the NAN character entropy feature, which represents the distribution of NAN characters in a URL, which substantially improves our classification accuracy.
In addition to these perspectives, we also tested Random Forest classifier with hyperparameter tuning. With NAN character entropy, we achieved a ROC AUC 96.20% of on the balanced dataset and 89.31% on the imbalanced dataset, outperforming the 87.51% and 84.31% obtained without NAN character entropy, respectively, i.e., 5 to 6% improvement. We proposed NAN character entropy to detect phishing websites only by analyzing URLs whenever little or no information is available in a webpage of the phishing website.
However, we still have a problem of high FPR. The limitation of the feature perspective is that we used DMOZ data, in which, with respect to the number of subdomains and path information, legitimate websites have no enough path information at all. (for example: https://www.facebook.com/). In contrast, phishing sites provide more path information (for example: http://67.205.147.248/oft/index.php?produto=722415036). It makes it difficult to retrieve similar patterns of path information for legitimate URLs, leading to high FPR.

6 Conclusion

Phishing detection has become a crucial research area as the number of phishing attacks grows along with e-commerce and the Internet transactions. We consider the URL to be a significant factor in preventing attacks because phishers create fake websites with as little information as possible in a webpage. To overcome the inability to retrieve much or any information about a webpage for detection, a URL-based approach is one solution. Since we consider that NAN character distribution is distinctive of phishing URLs, our contribution is to propose an entropy feature for NAN characters and compare it with previously proposed features. We found that combining our proposed feature with previous features outperforms previous features alone with respect to accuracy. Our system achieved ROC AUC scores of 89.31% and 96.20% on imbalanced and balanced datasets, respectively compared to the previous features only with 84.31% on imbalanced and 87.51% on balanced datasets. Moreover, the execution time is faster with our entropy feature than the one without entropy.
However, our approach still has a problem with relatively high with FPR (over 20%), which is our future work to decrease FPR.

REFERENCES

[1] B.B. Gupta, Nalin Asanka Gamagedara Arachchilage and Kostantinos E. Psannis. 2017. Defending against phishing attacks: taxonomy of methods, current issues and future directions. arXiv:1705.09819
[2] Kang Lang Chiew, Kelvin Yong and Choon Lin Tan. 2018. A survey of phishing attacks: their types, vectors and technical approaches. Expert Systems with

Applications, ScienceDirect 106 (Sept. 2018), 1-20. DOI: https://doi.org/10.1016/j.eswa.2018.03. 050

[3] Mahmoud Khonji, Youssef Iraqi and Andrew Jones. 2013. Phishing detection: a literature survey. Communications Surveys & Tutorials. IEEE 15, 4 (2013), 2091-2021. DOI:http://doi.org/10.1109/SURV.2013.032213.00009

[4] Gaurav Varshney and Manoj Misra and Pradeep K. Atrey. 2016. A survey and classification of web phishing detection schemes. Security and Communication Networks. Wiley Online Library 9, 18 (Oct. 2016), 6266-6284. DOI:http://doi.org/10.1002/sec.1674

[5] Doyan Sahoo, Chenghao Liu and Steven C.H. Hoi. 2017. Malicious url detection using machine learning: A survey. arXiv:1701.07179

[6] Adam Oest, Yeganeh Safei, Adam Doupe, Gail-Joon Ahn, Brad Wardman and Gary Warner. 2018. Inside a phisher's mind: understanding the anti-phishing eco system through phishing kit analysis. In Proceedings of 2018 APWG Symposium on Electronic Crime Research (eCrime '18). IEEE, 1-12. DOI:https://doi.org/10.1109/ECRIME.2018.8376206

[7] Ye Cao, Weili Han and Yueran Le. 2008. Anti-phishing based on automated individual white-list. In *Proceedings of the 4th. ACM Workshop on Digital Identity Management (DIM '08)*. ACM Press, New York, NY, 51–60. DOI:https://doi.org/10.1145/1456424.1456434

[8] Neil Chou, Robert Ledesma, Yuka Teraguchi and John C. Mitchell. 2004. Client-side defense against web-based identity theft. In *Proceedings of the Network and Distributed System Security Symposium (NDSS '04)*. DBLP Press, San Diego, California.

[9] Google. 2019. What are the Safe Browsing APIs? Retrieved from *https://developers.google.com/safe-browsing/v4/*

[10] Yogesh Joshi, Samir Saklikar and Debabrata Das and Subir Saha. 2008. PhishGuard: A browser plugin for protection from phishing. In *Proceedings of the 2nd. International Conference on Internet Multimedia Services Architecture and Applications (IMSAA '08)*. IEEE, 1-6. DOI:10.1109/IMSAA.2008.4753929

[11] Eric Medvet, Engin Kirda and Christopher Kruegel. 2008. Visual-similarity-based phishing detection. In *Proceedings of the 4th. international conference on Security and privacy in communication networks (SecureComm '08)*. ACM Press, New York, NY, Article 22, 1-6. DOI:https://doi.org/10.1145/1460877.1460905

[12] Engin Kirda and Christopher Kruegel. 2005. Protecting users against phishing attacks with AntiPhish. In *Proceedings of the 29th.International Computer Software and Applications Conference (COMPSAC '05)*. IEEE 2, 517-524. DOI:https://doi.org/10.1109/COMPSAC.2005.126

[13] Huaping Yuan, Xu Chen, S. Feng, Yukun Li, Zhenguo Yang and Wenyin Liu. 2018. Detecting phishing websites and targets based on URLs and webpage links. In Proceedings of the 24th. International Conference on Pattern Recognition (ICPR '18). IEEE, 3669-3674. DOI:https://doi.org/10.1109/ICPR.2018.8546262

[14] Dharmaraj R. Patil and Jayantro B. Patil. 2018. Malicious URLs detection using decision tree classifiers and majority voting technique. Cybernetics and Information Technologies. SCIENDO 18, 1 (Mar. 2018), 11-29. DOI: 10.2478/cait-2018-0002

[15] Alejandrp C. Bahnsen, Eduardo C. Bohorquez, Sergio Villegas, Javier Vargas and Fabio A. González. 2017. Classifying phishing URLs using recurrent neural networks. In *Proceedings of the* 2017 APWG Symposium on Electronic Crime Research (eCrime '17). IEEE, 1-8. DOI:https://doi.org/10.1109/ECRIME.2017.7945048

[16] Phishlabs. 2018. Hacking the human. Phishing trends and intelligence report.

[17] Free Hosting Sites For Phishers. 2012. Retrieved from https://blackbackhacker.blogspot.com/2012/09/free-hosting-sites-for-phishers.html

[18] PhishTank. Out of the Net into the Tank. Retrieved from https://www.phishtank.com/index.php

[19] DMOZ. The Directory of Web. Retrieved from https://web.archive.org/web/20170317132728/http://rdf.domz.org/rdf/