PhishX: An Empirical Approach to Phishing Detection

Jay Sinha*
jay.x.sinha@gmail.com
Ramaiah Institute of Technology
Bengaluru, India

Madhurendra Sachan*
madhurendra@tikaj.com
TIKAJ
New Delhi, India

ABSTRACT

Phishing techniques and attacks have been prevalent recently after they were first proposed in the 1980s. Yearly unique phishing attacks exceeded 1 million in 2015 and have been on the rise ever since. With the introduction of money-spinning ransomware attacks, phishing has become much more lucrative in order to make the first breach. Technology companies offer products ranging from blacklists to heuristics which can often prove ineffective when pinned against semantics-based attack structures. In this paper, we take an empirical approach towards phishing detection by proposing a data set taking into account 198 features extracted from more than 73,000 phishing websites. All the features taken into account don't require human intervention and are fully automated to help capture vast distribution among phishing URLs and their distinct combination of features. We also provide detailed analysis using Machine Learning and Deep Learning models, out of which Random Forest makes 93.09% accurate detection of phishing pages with an FPR of 0.122 when paired with the same features extracted from random 52,000 websites from the top Alexa list.

KEYWORDS

Phishing, Cybersecurity, Machine Learning, Deep Learning, Random Forests, Decision Trees, Gradient Boosting, Neural Networks

1 INTRODUCTION

Social engineering techniques making use of authority, intimidation, consensus, and urgency have long exploited vulnerabilities within organisations. Phishing is a type of social engineering technique designed to trick a human being into revealing sensitive information [29]. The word "Phishing" is a leetspeak variant of fishing with "ph" being a common replacement for "f" to lure users to "fish" for users' sensitive information. Phishing has resulted into data breaches like Sony Pictures [30], iCloud [28], US [10] and Ukranian Power Grid attacks [27]. Over the past three decades, starting from 1995's AOHell attack [1], phishing attacks have become extremely sophisticated. Attackers normally mirror the target website to track every action the user is making. FBI has labelled phishing attacks as the most common attack performed by cyber-criminals. In order to prevent phishing attacks, organisations provide training to their employees and partner with cyber security firms. Additionally, an active database of phishing URLs is provided by websites like OpenPhish [2], PhishStats [4], PhishBank [3] etc. Google provides a service called Safe Browsing [15] for this which helps identifies dangerous URLs as well. Microsoft Outlook (being the most used enterprise mailbox) has in-built tools and extensions for users to identify potentially dangerous emails and allow IT administrators to manage incoming traffic as well to overall reduce the chances of a successful phishing attack.

However, with the constant evolution of phishing techniques, all of above methods still do not allow to effectively preclude an attacker from targeting human vulnerabilities. In Reinheimer, Kunz, Volkamer and Renaud et al. [20], researchers highlight poor user knowledge and lack of browsing hygiene as the key factor of successful phishing attacks conducted in enterprises. Public service organisations like Anti Phishing Working Group [8] reported 245,000+ phishing attacks in January 2021 alone. Previous research on automated phishing based systems have taken into account black listing [24], TF-IDF analysis based on the content of page [31], visual features to compare similarity [7], using NLP to check URL itself [12], and search engine techniques in order to check authenticity of a webpage [11] [17]. The aforementioned techniques do not take into account features like type of CMS used, web server the website is running, presence of tag management system etc. which have a significant impact on the genuineness of a website or likelihood of detection of a zero-day attack. In this paper, we will explore an exhaustive list of extractable features in order to create our dataset on roughly 73,000 records of phishing URLs taken from various sources like OpenPhish, PhishBank etc. We will also provide insights into similar approaches that have been taken before to present delineating advantages which significantly helps improve the process of detecting phishing websites.

2 RELATED WORK

Phishing Detection systems have existed since phishing came into surface in enterprises, there have been a multitude of approaches to detect phishing attacks to protect users. A very well summarized survey has been published just recently which provides a list of different approaches and their conclusive results [6].

We have largely looked at deriving practical approaches of phishing classification which can scale with verification requests while providing acceptable accuracy with the least FPR. Though theoretically, significant number of researches discussed so far have produced 90% and above in prediction but very few of them have produced any practical example for the same. We tested the model against a general user's practical browsing history which yielded that up to 3% FPR which can be improved, considering the research will be used by PhishX.

Papers with higher accuracy have had a dependence on one of the list-based features like brands list, IP blacklists, etc., or high computation features which normally create a bottleneck in detection; being the slowest evolving component of the detection mechanism for fact that they are somewhat manually maintained

*Both authors contributed equally to this research.

or are limited by data sources. We have focused on features that don't require human intervention to eliminate bottlenecks.

Additionally, using list-based features in dataset features means introducing a highly positive feature to the algorithm which can create a bias towards one specific feature resulting in a biased algorithm model.

2.1 Phishing Feeds based detection

These approaches contain a URL blacklist which can be used to validate any URL for phishing attacks. Though these blacklists can sometimes be extended to block the domains but they depend on user-reported data and have limited capability to detect phishing. They largely depend on contributors reaction time which is only effective for attacks which have a long life cycle, while most of the phishing attacks are designed to harvest data within few days or even hours of spamming or attack. Also, these approaches need multiple sources to verify the data. Example of such approaches are Google Safe [15], PhishTank etc. There have been proposed work to fuzzy search blacklists like [22] which do improve the approaches but limitations still exist because the data is user initiated.

2.2 Webpage Link Features

These approaches analyze the links provided by the user, they don't require fetching the website content and running analysis, such approaches tend to be faster but are highly inaccurate in nature. We analyzed a batch of sample phishing URLs with a genuine set of URLs through statistical analysis producing no correlation between phishing and normal pages. The statistical analysis didn't pursue advanced features like brand presence, redirection, and discovery-oriented phishing attacks which always produce a relationship between phishing pages. We did consider a few link analysis features but they didn't compose in the primary feature set which had a significant impact in results.

Moreover, these methods also do not take into account presence of special-purpose TLDs like: .engineering, .dev, .support, .cx, .tech, .vc etc. and can very well tag legitimate websites with these TLDs in association with most-abused TLDs like: .shop, .work, .gq, .cam etc. [25]. Research conducted by [12] does not take into account the limitation of collection of legitimate brand names in order to cross verify them over to maintain an exhaustive list that takes into account all brand names. Moreover, these approaches can be very well targeted by making use of a root-level domain.

2.3 Webpage Features

Webpage features analysis requires analysis of the content of webpage which can only be done once a web request has been performed to the URL which requires network availability and webpage to be active while it is being accessed. The approach also has significant latency in processing the verdict due to reliance on real-time network dependency.

This approach is also prone to page redirection verification where attackers verify the redirection source of webpage and geo-blocking where an attacker can restrict access to few specific geographical location.

A lot of proposed method use webpage content analysis where the research involves extracting few HTML attributes of phishing page like wordlist, features of links present in pages, page structural features etc. [21] Webpage features analysis do strikes a balance between performance and accuracy, in certain cases achieving accuracy upto 99.2%.

2.4 Webpage Visual Features

Webpage visual analysis inherits restriction of webpage feature analysis and resources constraint with a promise to generate better accuracy. These utilized visual analysis of web pages which can be very effective as phishing attacks mostly tend to create a visual copy of webpages. They can use different approaches like obfuscation, image maps, iframes and other methods to hide the actual content/attributes of pages, But these approaches are very slow in nature and tend to consume much more resources. While accuracy is largely dependent on database against which visual similarity is matched. As executed by visual histogram analysis [18] [16], the approach requires fetching of page content and visual similarity analysis (which is a resource-intensive process) and doesn't work well for zero-day webpages. There have been other multitude approaches that use CSS & image-based visual similarity which can produce high accuracy results up to 97.30% but at the same time, require a strong dataset of known websites.

These web page-based features when combined with Machine Learning or Deep Learning models, give results ranging from low 90s to around 99% test set accuracy while keeping FPR in the range of 0.01 to 0.2. While the result metrics are themselves appreciable, the analysis only takes into account less than 10,000 phishing records while considering at max 50 top-level features which make for a small dataset that can be very well tagged as unrepresentative of the vast and complex underlying nature of phishing websites [14]. Models trained on these shallow datasets can prove to be brittle when exposed to new and more sophisticated phishing websites [5]. While statistical methods like sampling can help in combating this small dataset problem, it does not make the model effective against anomalies resulting from exposure to new data.

3 DATASET

The dataset was being progressively stored into a MongoDB collection which allowed us to experiment with features without the need to pre-define the list of all the features. We used URLs from OpenPhish for collecting verified phishing pages for the phishing dataset and took into account random 52,000 Alexa websites from the top 1 million for the non-phishing dataset. The high-level process of collecting phishing pages had the following pointers:

- The OpenPhish data was live and required immediate fetching of webpages before they were removed so we monitored OpenPhish feed for new URLs every 12 hours. (limited by free feed publish rate)
- Verify if URL has been scraped; if it was skipped.
- Try connecting with the URL with 10 seconds timeout.
- In case, URL's HTML still cannot be fetched, record the exception as result but don't mark website as scraped. This can happen where URL was removed before it was published or timeout occured.
- Retry fetching all failed websites once in a day.

Table 1: Top level features

Column Name/Prefix	Description
Analytics	Website analytics integration for error, CRM, ads, testing etc.
CDN	CDN is being utilized for content delivery
CMS	Type of CMS used for managing the website
Web Master	Website has webmaster registration keys
Web Server	Which type of webserver is being used
Ads	Ads technologies like ad analytics, ad exchange trackers used
Copyright	Copyright symbol or restriction present
Current year copyright	if copyright is current year
External Sites	Number of links to external sites
Domains	Number of unique domains present in content
Feeds	If content on website is copied from another website (maybe using Syndication techniques)
Framework	Any framework used to build website
home_main_ngram_intersection	Webpage content intersection with provided page
hosting	Type of hosting where website is hosted
Javascript	Javascript library is used
language	Language of content
links	The number or URLs present in the HTML of phishing page
mapping	Mapping integration
media	Media integration like YouTube, Sound Cloud etc. present
mobile	Mobile Related optimization
mx	Mail Server types
nDescription	Length of meta description
nTitle	Length of meta title
ns	Name Server related information
parked	if domain is parked
payment	Payment providers integration
privacy_policy	Privacy policy present
robots	Robots config is provided
shipping	Shipping providers are configured
shop	E-commerce store
ssl	SSL Certificate configurations
widgets	Extensions that are used by CMS

- To bypass simple bots validation, we used random User Agents headers and other relevant headers to fake browser use while requesting pages.
- We collected few features like root domain page similarity with phishing page, CMS used etc. which we decided during preliminary analysis required the website to be live.

These requests were performed from a pool of few Random IPs to bypass any IP blocking or geo-restrictions applied, the servers were located in New Jersey, Mumbai and London. We didn't validate any URLs for geo-restrictions which was possible in highly sophisticated phishing attacks. Collecting data for Alexa 52,000 has a similar process as mentioned above except we didn't have to monitor any sources for new URLs. Post collection of HTML of each URL, we improvised & experimented with additional features, where analysis was executed on page source collected earlier.
The features which have been extracted from the dataset have been summarized in **Table 1** at a very high level, while their subcategories suffixed with an underscore (_) highlights their classification. For e.g.-hosting_australian-hosting tells that the hosting of the website is on a VPS based in Australia.

Below are some non-features columns in the dataset which are not being used in the final part of the analysis:

- Status indicates if page has been crawled
- Alexa, if page is part of Alexa websites
- HTML if string it is HTML content, if it is an object page had errors during fetch
- Title contains the text in title tag of the HTML
- Description contains the main meta tag's description

All the features collected are automated with the scraping script and do not require any human intervention in order to provide manual features. Although, we considered including safe_browsing as a feature that would indicate it being flagged by Google Safe Browsing; we skipped it since OpenPhish can be indexed by crawlers put in place for Google Search which would make this feature a ground truth in the majority of phishing entries. Moreover, features like

framework, analytics, CDN have been added in order to measure the extent to which the systems in place are tracking user actions. In simpler cases, detection of these three features will tell that the website is a phishing website or not.
The scraping script helps put in place a system that will keep on extracting a vast number of features even in the future to track the similarities and dissimilarities between phishing website metrics that will help track the evolution of phishing websites in general to detect and visualise the level of sophistication used in order to trick users.

4 ANALYSIS

For the analysis part of this paper, we have taken the same features extracted from 52,000 Alexa websites as we have for phishing websites. We will discuss the approach taken for the analysis part which includes: preprocessing, models, evaluation, and finally, results. Details on the exact dataset and Jupyter notebooks for the purposes of the reproduction will be covered under the Future Work section. Our analysis is based to evolve with the attack mechanism, though we haven't executed a feedback training model, the features used are non-conventional in nature and directly relate to the cost of conducting a phishing attack. We manually analyzed 100's of phishing websites in order to create a scraping script that extracts all the features from a website.
Following are some points to be noted based on which we created a list of features that can be evolved with time:

- Phishing websites tend to be hosted on compromised websites, as the cost of attack is low.
- Attackers don't own the domain i.e. the assets and other content of websites have different origin from the current website or root domain has difference in nature of content.
- Attackers don't care about SEO & other such details, they only focus on visual similarity and so they end up ignoring tags, sitemap, etc.
- Phishing pages have strong relations with outbound links, wordlist and other webpages attributes, which has been highlighted in other researches as well.
- They mimic a brand's presence using visual hints, which are sometimes embedded using images, pure CSS, iframes, javascript and other approaches to bypass HTML-DOM detection.

For the purpose of this analysis, we have entirely ignored all the textual information related to each phishing record: URL, title, Description and Server etc. This approach is taken in order to force the models to learn to identify phishing websites solely on the basis of metrics that consider website content, website domain, their technical specifications like hosting, CMS, widgets etc.

4.1 Preprocessing

While appending 52,000 Alexa websites to the dataset, an additional Boolean column called 'phishing' has been introduced which when true tells that it is a phishing record. This will function as our result column. For preprocessing, the columns (description, title, URL, and Server) have been removed. Columns having non-binary numerical values have been normalised. Dataset is shuffled and then 80% is provided as training while remaining 20% is allocated as test set for classifier models. In case of Neural Networks, 80% is taken as training set whereas 10% each is allocated to validation and test set.

Table 2: Analysed Models with Test Accuracy and FPR

Model	Accuracy	FPR
Decision Tree	89.59%	0.1520
Random Forest	93.09%	0.1222
Gaussian Naive Bayes	80.42%	0.3665
KNN	91.04%	0.1520
XGBoost	92.89%	0.1292
CatBoost	92.84%	0.1255
SVM	87.73%	0.1978
ANN	92.64%	0.1293
1D-CNN & BiLSTM	92.87%	0.1272

4.2 Models

We have run analysis using following classifiers: Decision Tree, Random Forest, Gaussian Naive Bayes, KNN, SVM, XGBoost and CatBoost using binary classification.
We have also run analysis using following Neural Networks: ANN, & 1D-CNN & BiLSTM. While all the classifiers and basic Neural Networks are self-explanatory for their usage in a binary classification setting, a hybrid and hierarchical Neural Network like 1-D CNN & BiLSTM provides robust learning (when paired with overfitting preventing techniques like the inclusion of a Dropout Layer) by combining spatial learning and temporal learning of BiLSTM and CNN layers respectively [13]. Neural Networks are trained using Adam optimizer with categorical_crossentropy loss.
Model performance will be evaluated using test-set accuracy and FPR.

4.3 Results

Test set Accuracy and FPR metrics can be found for the models outlined in previous section in **Table 2**.

So far, Random Forest [9] classifier gives the best performance on which we have evaluated Feature Importance analysis using MDI and Permutation Importance as well.

As per MDI [26] analysis, features: cdn, analytics, analytics_audience-measurement, Web Server, ssl, analytics_visitor-count-tracking, analytics_application-performance, javascript, widgets, domains are top 10 most important features. Top 20 features contributing to the analysis are in **Figure 2** and it can be clearly seen that cdn, analytics, ssl and framework contribute heavily to identifying phishing websites in our dataset.

Since, MDI analysis can be biased on datasets having abundance of unique features [23], we ran Permutation Importance analysis too. The top 20 features recognised (in **Figure 3**) include mx, Web Master, links, widgets, analytics, ads, ssl, ns, cdn, CMS. It is evident that 16 features in both analysis are same which are listed in the **Table 3** and signify their importance towards contributing to identification of phishing websites.
As per **Figure 1**, FPR of 0.1222 is observed which means that about

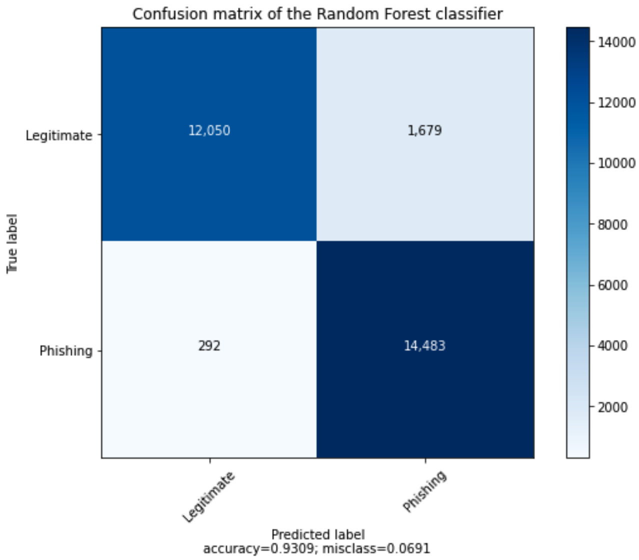

Figure 1: Confusion Matrix of Random Forest Classifier

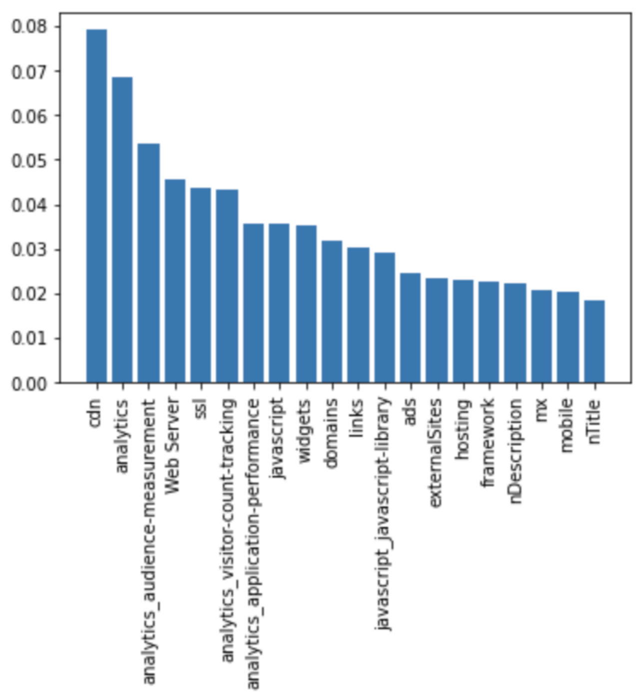

Figure 2: Important Features identified using MDI Analysis

12% of the times, the classifier is tagging a legitimate website as a phishing website and could be further improvement down the line.

Table 3: 16 Common Important Features identified by MDI and Permutation Importance Analysis

mx	analytics
Web Server	ssl
javascript	widgets
domains	links
ads	externalSites
hosting	framework
nDescription	mobile
nTitle	cdn

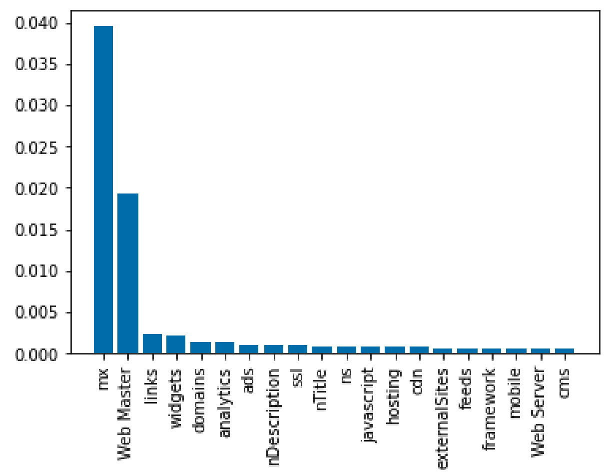

Figure 3: Important Features identified using Permutation Importance Analysis

5 CONCLUSION

We proposed an enhanced and effective approach towards detecting a phishing website by taking a holistic approach towards the extraction of features that provide a lot more data than usual visual features or information extracted from the URL alone. We also identified the contribution of root domain features, cdn, meta tags, analytics and generic HTML-based features that contributed towards accurately predicting whether the given website is a phishing website or not.

This is a first-of-its-kind hybrid approach that doesn't just look at a page to conclude if it is a phishing page but also helps in identifying the likelihood of a domain/website being compromised for phishing based on certain features like CMS, hosting, CDN etc. which historically, have significant contribution in mass phishing hosting.

6 FUTURE WORK

We can further improve the accuracy using visual classification as used by other researchers and improvise to detect different brands. FPR obtained in the analysis is also something that needs to be improved in the future. As of now, our data captures most of the metrics that can be easily obtained and processed, however, there

is still a need of NLP techniques in order to parse the content and cross-verify legitimacy in languages other than English.
The dataset used in this paper will be published on PhishX which will primarily offer a real-time feed of phishing websites collated from different reporting platforms. It will also provide all the features discussed in this paper corresponding to each individual URL. All the data will be available for free and will be maintained by the authors themselves as an open-source initiative under Creative Commons Attribution 4.0 International License [19]. The main purpose for providing real-time data is to provide data on scale to boost the amount of independent experiments and research for more efficient automated phishing detection systems in the future.

REFERENCES

[1] [n.d.]. *History of Phishing.* https://www.phishing.org/history-of-phishing
[2] [n.d.]. *OpenPhish.* https://openphish.com
[3] [n.d.]. *PhishBank.* https://phishbank.org/#/
[4] [n.d.]. *PhishStats.* https://phishstats.info/
[5] 2017. *Artificial Intelligence and Machine Learning Applied to Cybersecurity.* Technical Report. https://www.ieee.org/content/dam/ieee-org/ieee/web/org/about/industry/ieee_confluence_report.pdf
[6] X. Liu A.R. Javed Z. Jalil K. Kifayat A. Basit, M. Zafar. 2020. A comprehensive survey of AI-enabled phishing attacks detection techniques. Telecommunication Systems (2020). https://link.springer.com/article/10.1007%2Fs11235-020-00733-2
[7] B.B. Gupta A.K. Jain. 2017. Phishing detection: Analysis of visual similarity based approaches. In *Security and Communication Networks.* Hindawi.
[8] APWG. [n.d.]. *Anti Phishing Working Group.* https://apwg.org/
[9] Leo Breiman. 2001. Random Forests. *Machine Learning* (2001). https://doi.org/10.1023/A:1010933404324
[10] CISA. [n.d.]. *Russian Government Cyber Activity Targeting Energy and Other Critical Infrastructure Sectors.* https://us-cert.cisa.gov/ncas/alerts/TA18-074A
[11] K. Wong S.N. Sze C.L. Tan, K.L. Chiew. 2016. PhishWHO: Phishing webpage detection via identity keywords extraction and target domain name finder. In *Decision Support Systems.* Science Direct.
[12] O.K. Sahingoz O. Demir E. Buber, B. Diri. 2019. Machine learning based phishing detection from URLs. In *Expert Systems with Applications.* Research Gate.
[13] Y. Chen W. Che G. Xu, T. Ren. 2020. A One-Dimensional CNN-LSTM Model for Epileptic Seizure Recognition Using EEG Signal Analysis. In *Advanced Deep-Transfer-Leveraged Studies on Brain-Computer Interfacing.* frontiersin. https://doi.org/10.3389/fnins.2020.578126
[14] J. Luo GJ Qi. 2021. Small Data Challenges in Big Data Era: A Survey of Recent Progress on Unsupervised and Semi-Supervised Methods. (2021). https://arxiv.org/pdf/1903.11260.pdf
[15] Google. [n.d.]. *Google Safe Browsing.* https://safebrowsing.google.com
[16] S.C. Wang K.T. Chen I.F. Lam, W.C. Xiao. 2009. Counteracting Phishing Page Polymorphism: An Image Layout Analysis Approach. In *Advances in Information Security and Assurance.* Springer.
[17] W.K. Tiong K.L. Chiew, E.H. Chang. 2015. Utilisation of website logo for phishing detection. In *Computers Security.* Science Direct.
[18] C.R. Huang C.S. Chen K.T. Chen, J.Y. Chen. 2009. Fighting Phishing with Discriminative Keypoint Features. In *IEEE Internet Computing.* IEEE.
[19] TLDR Legal. [n.d.]. *Creative Commons Attribution 4.0 International.* https://tldrlegal.com/license/creative-commons-attribution-4.0-international-(cc-by-4)
[20] B. Reinheimer A. Kunz M. Volkamer, K. Renaud. 2017. User experiences of TORPEDO: Tooltip-poweRed Phishing Email Detection. In *Computers Security.* Science Direct.
[21] Y. Yuan N. Zhang. 2012. Phishing Detection Using Neural Network. (2012). http://cs229.stanford.edu/proj2012/ZhangYuan-PhishingDetectionUsingNeuralNetwork.pdf
[22] R.R. Kompella M. Gupta P. Prakash, M. Kumar. 2010. PhishNet: Predictive Blacklisting to Detect Phishing Attacks. In *IEEE Annual Joint Conference: INFOCOM, IEEE Computer and Communications Societies.* IEEE.
[23] R. Hornung S. Janitza. 2018. On the overestimation of random forest's out-of-bag error. In *PLOS ONE.* PLOS.
[24] G. Warner L. Cranor J. I. Hong C. Zhang S. Sheng, B. Wardman. 2009. An Empirical Analysis of Phishing Blacklists. In *Proceedings of the 6th Conference on Email and Anti-Spam.* CMU.
[25] Spamhaus. [n.d.]. *The 10 Most Abused Top Level Domains.* https://www.spamhaus.org/statistics/tlds/
[26] J. Friedman T. Hastie, R. Tibshirani. 2008. The Elements of Statistical Learning. (2008). https://web.stanford.edu/~hastie/Papers/ESLII.pdf#page=611

Table 4: Abbreviations and their full forms

Abbreviation	Full Form
ANN	Artificial Neural Networks
APWG	Anti Phishing Working Group
CatBoost	Category Boosting
CDN	Content Delivery Network
CMS	Content Management System
CNN	Convolutional Networks
CRM	Customer Relationship Management
CSS	Cascading Style Sheets
FPR	False Positive Rate
HTML	Hyper Text Markup Language
IP	Internet Protocol
KNN	K-Nearest Neighbors
LSTM	Long Short Term Memory
MDI	Mean Decrease in Impurity
MX	Mail Exchange
NLP	Natural Language Processing
NS	Nameserver
SEO	Search Engine Optimization
SSL	Secure Sockets Layer
SVM	Support Vector Machine
TF-IDF	Term Frequency–Inverse Document Frequency
TLD	Top level Domain
URL	Uniform Resource Locator
XGBoost	eXtreme Gradient Boosting

[27] Wikipedia. [n.d.]. *December 2015 Ukraine power grid cyberattack.* https://en.wikipedia.org/wiki/December_2015_Ukraine_power_grid_cyberattack#:~:text=On%2023%20December%202015%2C%20hackers,cyberattack%20on%20a%20power%20grid.
[28] Wikipedia. [n.d.]. *iCloud leaks of celebrity photos.* https://en.wikipedia.org/wiki/ICloud_leaks_of_celebrity_photos
[29] Wikipedia. [n.d.]. *Phishing.* https://en.wikipedia.org/wiki/Phishing
[30] Wikipedia. [n.d.]. *Sony Pictures Hack.* https://en.wikipedia.org/wiki/Sony_Pictures_hack
[31] L. F. Cranor Y. Zhang, J. I. Hong. 2007. Machine learning based phishing detection from URLs. In *16th international conference on World Wide Web.* ACM.

A ABBREVIATIONS

B COMPLETE FEATURES LIST

Table 5: Complete Features List

URL	Title	Description
Web Master	Web Server	ads
ads_ad-analytics	ads_ad-blocking	ads_ad-exchange
ads_ad-network	ads_ad-server	ads_ads-txt
ads_adult	ads_affiliate-programs	ads_audience-targeting
ads_content-curation	ads_contextual-advertising	ads_data-management-platform
ads_demand-side-platform	ads_digital-video-ads	ads_dynamic-creative-optimization
ads_fraud-prevention	ads_mobile	ads_multi-channel
ads_retargeting-/-remarketing	ads_search	analytics
analytics_a/b-testing	analytics_advertiser-tracking	analytics_application-performance
analytics_audience-measurement	analytics_cart-abandonment	analytics_conversion-optimization
analytics_conversion-tracking	analytics_crm	analytics_data-management-platform
analytics_error-tracking	analytics_feedback-forms-and-surveys	analytics_fraud-prevention
analytics_lead-generation	analytics_marketing-automation	analytics_mobile
analytics_personalization	analytics_product-recommendations	analytics_retargeting-/-remarketing
analytics_site-optimization	analytics_social-management	analytics_tag-management
analytics_visitor-count-tracking	cdn	cdns
cdns_edge-delivery-network	cms	cms_agency
cms_automotive	cms_blog	cms_community-cms
cms_ecommerce-enabled	cms_enterprise	cms_financial
cms_forum-software	cms_headless	cms_healthcare
cms_hosted-solution	cms_job-board	cms_landing-page
cms_learning-management-system	cms_non-profit	cms_open-source
cms_real-estate	cms_simple-website-builder	cms_social-management
cms_ticketing-system	cms_wiki	copyright
copyright_presence	current_year_match_copyright	domains
externalSites	feeds	framework
framework_schema	framework_wordpress-theme	home_main_ngram_intersection
hosting	hosting_australian-hosting	hosting_canadian-hosting
hosting_chinese-hosting	hosting_cloud-hosting	hosting_cloud-paas
hosting_dedicated-hosting	hosting_dutch-hosting	hosting_german-hosting
hosting_hong-kong-hosting	hosting_japan-hosting	hosting_shared-hosting
hosting_uk-hosting	hosting_us-hosting	hosting_vps-hosting
javascript	javascript_animation	javascript_charting
javascript_compatibility	javascript_framework	javascript_javascript-library
javascript_jquery-plugin	javascript_slider	javascript_ui
language	link	link_adult
links	mapping	mapping_maps
media	media_digital-video-ads	media_enterprise
media_live-stream-/-webcast	media_online-video-platform	media_social-video-platform
media_video-analytics	media_video-players	mobile
mx	mx_business-email-hosting	mx_campaign-management
mx_dmarc	mx_marketing-platform	mx_secure-email
mx_transactional-email	mx_web-hosting-provider-email	nDescription
nDescriptionTitle	nTitle	ns
ns_enterprise-dns	ns_tld-redirects	parked
payment	payment_bitcoin	payment_checkout-buttons
payment_currency	payment_donation	payment_pay-later
payment_payment-acceptance	payment_payments-processor	privacy_policy
robots	shipping	shop
shop_enterprise	shop_hosted-solution	shop_multi-channel
shop_non-platform	shop_open-source	shop_plugin-/-module
ssl	ssl_extended-validation	ssl_root-authority
ssl_wildcard	widgets	widgets_bookings
widgets_bookmarking	widgets_call-tracking	widgets_captcha
widgets_charting	widgets_comment-system	widgets_content-modification
widgets_ecommerce	widgets_error-tracking	widgets_feedback-forms-and-surveys
widgets_financial	widgets_fonts	widgets_image-provider
widgets_live-chat	widgets_login	widgets_marketing-automation
widgets_mobile	widgets_privacy-compliance	widgets_push-notifications
widgets_schedule-management	widgets_site-search	widgets_social-sharing
widgets_ssl-seals	widgets_tag-management	widgets_ticketing-system
widgets_tour-site-demo	widgets_translation	widgets_web-badge
widgets_wordpress-plugins	ads_bitcoin	hosting_french-hosting
hosting_italian-hosting	hosting_swiss-hosting	hosting_wordpress-hosting
shop_woocommerce-extension	shop_wordpress-plugins	widgets_joomla-module