

Phishing URL prediction – two-phase model using logistic regression and finite state automata

Nisha T N, Dhanya Pramod

Symbiosis Centre for Information Technology, Symbiosis International (Deemed University), Pune, India

Article Info

Article history:

Received Oct 25, 2024

Revised Mar 21, 2025

Accepted Jul 3, 2025

Keywords:

Attack probability detection

Feature selection

Finite state machine

Logistic regression

Malicious score

Phishing sites

Unintentional insider threats

ABSTRACT

The human factor in security is more important when they become the carriers of attacks on enterprises. Phishing attacks can be classified as insider attacks when the employees unintentionally participate in the attack propagation. Since complete user training is a myth, enterprises must implement detection tools for phishing attacks on their network perimeters. This research discusses a two-phase model for phishing URL detection, in which the first phase identifies the properties of URLs that detect phishing and their relative weight using logistic regression. The second phase checks the probability of a new URL being categorized as phishing using the knowledge achieved during the first phase using the dynamically created Finite state machines. The model defines a malicious score (MS), which can be used to check any URL in real-time to identify whether it is phishing or not. The model described in this work has been experimented with different benchmarking datasets to verify the performance. The model provided a decent result in classifying a URL as phishing or naive. The malicious score (MS) defined by this model can be used to evaluate any URL and can be used as a filtering mechanism for end-point phishing URL detection. The key contribution is towards developing a two-phase model which evaluates the URL with the help of self-crafted features without reliance on a feature set. This accommodates the model's hyper-competitive phishing URL detection area in cyber security.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Nisha T N

Symbiosis Centre for Information Technology, Symbiosis International (Deemed University)

Pune, India

Email: nisha@scit.edu

1. INTRODUCTION

Information security issues are increasing daily, regardless of the inventions happening in the security area. As the saying goes, "A chain is only as strong as the weakest link." The weakest link in information security is humans. Despite the robust security architecture and policies, organizations still experience breach because of the actions of humans involved in the information security architecture. For any organization, employees are considered to be the greatest asset. However, from a security perspective, they can be a liability to the company. Human actions, whether intentional or unintentional, give rise to security implications. As per the 2024 data security incident report by Baker Hostetler, security incidents have continued to be the leading in the market, and ransomware has been the cause for the last five years [1]. According to an IBM report, there has been a 71% increase in cyber threats, and in many cases, the attacks were initiated by utilizing human behaviour [2]. These phishing attacks account for most security incidents, which can be classified as unintentional threats, despite a small fraction of internal thefts, which can be considered intentional. One reason for this unintentional exploit of an organization's security posture is social

engineering attacks. Social engineering capitalizes on human psychology and deceives the victims to do the attack. Attacks are shifting from automated tools to social engineering attacks, with email being the most used tool [3].

Based on anti-phishing working group's (APWG) report on the phishing scene for the year 2024, phone-based phishing attacks are showing an all-time high trend and are going undetected. It shows a continuously increasing trend even in previous years, and for years, the number of reported phishing websites, emails, and targeted brands has risen steadily. APMG also reports that phishing attacks occur most frequently on the domains webmail, financial and payment sectors [4].

Creating security-aware users through training is the preeminent solution for phishing attack detection. As this aim is challenging to achieve, enterprises need to depend on the classification of phishing sites by blacklisting, heuristics, visual similarity or machine learning. Phishing detection by blacklisting requires the URL to be previously detected as phish, heuristics depend on the already present characteristics of the phishing URL and visual similarity detection is based on the content code.

Due to the availability of massive data sets of phishing and naïve URL databases, machine learning-based phishing detection methods are prominent in the area. Due to this reason, data mining and machine learning techniques are finding their importance in phishing URL detection, and models are constructed by taking advantage of different clustering algorithms.

The first layer of defence against phishing is achieved by identifying the context of phishing; basically, the email carrying phishing URLs to the victim's sight. Features of email are identified and modelled the classifiers using different machine learning techniques like SVM [5], WordNet ontology [6], multiple deep learning models [7], recurrent convolutional neural network model [8], TF-IDF based detection [9], deep learning model [10] are employed either as Signature-based or rule-based methods for the classification of phishing email [11], [12].

The URL to the malicious sites looks different than a normal URL. This idea is applied to URL-based phishing email detection. The lexical features of URLs are identified and used to detect phishing URLs. These lexical features are analyzed using different machine learning techniques such as SVM, random forest, Naïve Bayes, logistic regression, decision tree, confidence weighted algorithm, adaptive regulation of weights AROW K-means, neural networks, SOM, and compared the results [13]. URL optimal features other than these are extracted and applied to the frequent rule reduction (FRR) algorithm to detect phishing URLs [14]. Studies with multiple ML models and their enhancements are also proposed with high accuracy and efficiency [15], which does not necessitate a webpage visit [16].

Attackers obfuscate the URL using different techniques to avoid detection by analyzing lexical features. Lexical features combined with domain-based and content-based, thus provided good detection accuracy while using the same machine learning techniques [17]. Rule-based algorithms such as RIPPER, RISM, C4.5, CBA, and artificial neural networks are also used for phishing detection based on URLs and other features [18].

Other than lexical features, the differences between the visual link and actual link and misspelt or large host names are some of the unique features researchers identify. A two-phase model with a URL prediction component and an approximate URL matching component that matches the new URL with the blacklist is also developed [19]. Along with lexical and other features, some models combined fuzzy logic [20], some with blacklisted domains [21], and SHA1 hash and presence of login age [22] to detect phishing. These multi-stage detections also provided new models for phishing detection.

Content-based phishing detection is also employed, but is criticized for the danger of downloading the content for examination and the cost of time, bandwidth and resources. The anomalies in the web page by analyzing the content of the web page with a weighted TF-IDF model [23], signature for the page [24], MD5 hashes of the pages [25], login page features [26], keywords [27], images and scripts [28], [29].

Phishing URL detection techniques evolved using lexical, host-based, and content-based features and leveraging deep learning techniques [30], [31]. Deep learning models based on long short-term memory and deep neural networks are also employed successfully for phishing URL detection [32]. Techniques evaluating the word embeddings and character embeddings from the URL with a CNN-based URLNet [33] and deep learning-based Texception [34] also evolved.

This research uses URL-based classification as it will provide a good amount of predictability due to the availability of large numbers of phishing and Naïve URL databases, and can handle false negatives effectively. This model capitalises on the availability of a vast data set for identifying the features that truly classify the phishing URLs, confirming the features identified by the literature review, and is verified using logistic regression applied on different datasets. This model also relies on the knowledge-based state machine model as the probabilistic model to predict the URL as a malicious URL. This model is different from the state machine-based model suggested by Phish tester [35], where the behaviour of the webpage is evaluated using the request-response pair for each webpage component. The naive idea over here is that it does not directly depend on the input data set of phishing and naïve URLs. It uses the {feature, weight} tuple created

by the first stage in identifying the probability of a URL being phishing, as we are not opening the malicious link in this method unless the previous model reduces the chances of being infected. This will confirm the URL to be phishing by converging the effect of a primarily used feature and the probability generated by the model.

2. METHOD

URL-based detection is criticized for the possibility of attackers obfuscating the link to evade detection and the delay in blacklisting, which increases the false negatives in the detection. To address these gaps, this research proposes a two-phase model to detect phishing URLs by predicting the URL's probability of being malicious (Figure 1). The first component learns the structure of a phishing URL by implementing logistic regression on the features and training the classifier. It identifies the properties that truly classify a URL into naive and phishing using logistic regression, which calculates their relative ranks in detecting URLs. The classifier will be trained by both blacklist and white list URLs collected at different sources at different scopes. The second component then utilises this probability in identifying URLs using state machine-based evaluation. The model is tested against some known datasets, and the results are evaluated.

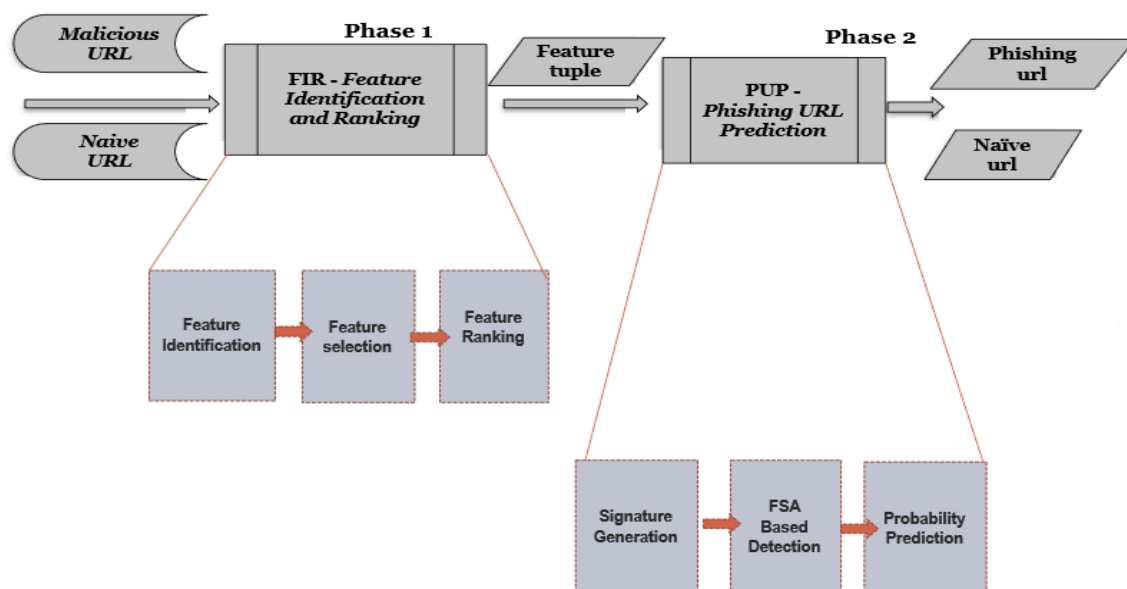


Figure 1. Two-phase phishing detection- general architecture

Logistic regression is a powerful and flexible model that demonstrates the probabilistic dependencies of the features involved in decision-making. Unlike the other machine learning models, logistic regression suffers the lowest False Positives, so it is preferred as false positives are more expensive than false negatives. However, if used independently, logistic regression is not the best fit for phishing detection, and it conflicts with other methods. The simplicity and interpretability of logistic regression justify the first stage of classification. The distinctiveness of this study is the use of logistic regression as a partial component in classification, other than using it as a method for it. Instead of classifying the URL only based on the logistic regression model trained on available datasets, we try to leverage the probability value generated as a feature weightage in abnormality prediction.

FSAs on the other hand, are excellent at establishing a sequential relationship between events and keeping track of activities over time. It guarantees that the current behaviour depends on all the previous events, and the dependency effectively predicts the pattern's linear behaviour. Combining the probabilistic logistic regression and linear FSA adds to the strengths of the two-phase model. This ensures the model works with temporal dependencies enhanced by a probability-based model.

2.1. Phase 1: feature identification and ranking (FIR)

As the literature review summarises, this phase helps the model choose the right features for the next knowledge-based phishing URL prediction (PUP) phase. The process moves through three steps: feature

identification, feature selection and feature ranking. The primary feature selection is based on the literature review output and then appraised using logistic regression.

Feature identification helps to determine the relative importance of the feature under consideration. The power of logistic regression in quantifying the relative effect of an independent variable on the dependent variable is used in this phase. This model does not depend on the extracted URL features list available online. We extracted the URL features from the URLs given and identified some features that successfully classify the phishing URL from the benign URL. To finalise the features and reinforce their relative importance with other features, they are checked against the standard datasets. The relative presence of these features is evaluated by their relative presence and their contribution towards classifying the URLs are studied.

The coefficient value in the logistic regression expresses the contribution of the particular feature in determining whether the URL is phishing or not. The odds ratio measures the likelihood of an event, and the probability value is derived from the odds ratio as:

$$prob = odds\ ratio / (1 + odds\ ratio) \quad (1)$$

2.2. Phase 2: phishing URL prediction (PUP)

This phase predicts a URL to be malicious by employing signature creation, FSA-based detection and the attack probability prediction. Signature creation utilizes the formal language model. A formal language $L1$ over a defined alphabet set Σ is an infinite set of strings defined over the alphabet Σ . Regular language can be expressed using a formula of Boolean logic, known as regular expressions. We define a regular language with 2 symbols, $\{1,0\}$, as a binary string of n positions:

$$L1 = \{x \in \{1,0\}^* / x \text{ is a string of size } n\} \quad (2)$$

The value n is the number of features used to evaluate whether the URL is phishing or not. The language defines the malicious URL as a string with at least one '1' in the string. A string with all '0's is the URL with no signs of malicious traces. The greater the frequency of '1's it has, the more probable the URL is malicious. Regular languages are encoded using finite state automata, or in other words, can be evaluated using finite state automata defined for that language. A finite state automata (deterministic finite automata) is defined as a five-tuple notation.

$$M = (Q, \Sigma, \delta, q_0, F) \quad (3)$$

Where Q denotes finite set of states, Σ denotes finite set of input symbols, δ denotes transition function, q_0 is the start state where $q_0 \in Q$ and F is the set of final or accepting states which is a subset of Q .

Based on the formal definition of finite state automata, PHISH_FSA, which evaluates the regular expression signature, is defined as given in Figure 2. The state machine with 9 states where state Q_0 is the initial state. The state machine classifies all strings ending in Q_1 as a safe URL and does not contain the features that define a malicious URL. Strings ending on any other state indicate the probability of the URL being malicious and the probability is calculated in the probability prediction phase.

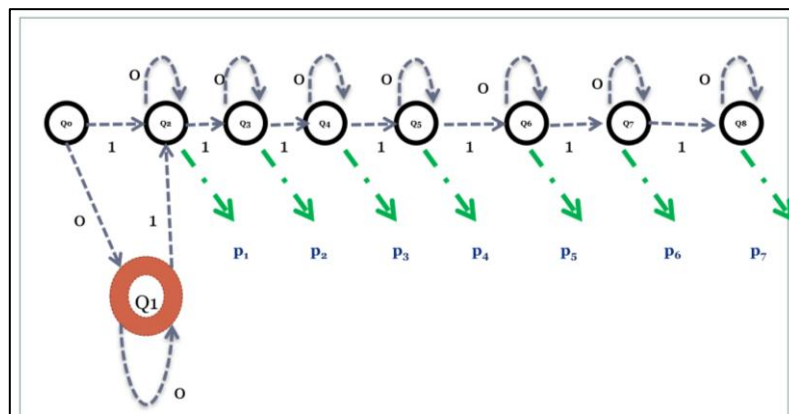


Figure 2. PHISH_FSA

The last step in this phase is probability prediction, where the probability of the URL being malicious is evaluated as MS. Malicious_Score is defined using two factors, present feature count (PFC) and probability value (PV). The formulation condenses the contributions of each feature as a function of its value, indicating the presence, position indicating the relevance, and probability indicating the contribution in prediction. Powering the position value with 2 will claim the conversion of the binary positional value to weightage. The mathematical representation of MS is displayed in (4) and (5).

$$MS = f(PFC, PV) \quad (4)$$

$$MS = \sum_{i=1}^n (VAL_i \times POW(2, POS_i) \times PV[POS_i]) \quad (5)$$

where VAL_i , is the value of regular expression at position i and POS_i , is the position of the present feature in the regular expression. Present feature count (PFC) refers to the number of features found present in the URL and its position in the regular expression, which classifies it to a malicious one and identifies the severity of the chance of the URL being blacklisted. Probability value (PV) is the feature probability array defined from the previous stage.

Calculating malicious scores uses two feature properties: feature position in the regular expression and the calculated probability from logistic regression. This empowers the prediction of malicious URLs by implanting the feature importance with probability and feature relevance with position. As the probabilities are calculated by analyzing a data set after identifying the popular features of the malicious URL, the false positives are reduced. Finally, the MS is calculated for each URL and is alerted with the score. The network admin can then utilize the score to block the URL from the network.

This detection's critical area is identifying the threshold value with which the MS can be benchmarked. Considering the flexible nature of the URL features, we decided to work with a flexible threshold value. The threshold value is calculated by evaluating different datasets available and agreeing on the MS score. The MS value calculated is evaluated to find the confusion matrix to evaluate the model's performance. This enables the model to be flexible enough to accommodate any future change in the feature evaluation. This necessitates a continuous fixation of threshold values by evaluating the recent dataset trends.

3. RESULTS AND DISCUSSION

3.1. Data set and experimental setup

The model requires data to be fed in two different phases. To include the variability in the URL data set, we have collected data from various sources, including Kaggle [33], PhishTank [34], and the Common crawl data set [35]. The selection of different datasets from different scopes convinces the model's reliability as the URL data is volatile and constantly changing. The data set provides both naïve and phishing URLs to train the model. Kaggle data sets are used to train the model, but only extract the raw URLs from the data set. Phishing data set is collected from Phishtank regularly, and naïve data from Common Crawl. A heterogeneous data set is generated by combining datasets collected from different data sources at different intervals. We used five training datasets with both positive and negative URLs uniformly allocated and five different datasets for testing, which included unary data. The model is developed by Python code, version 3.11.5, with standard libraries. The experiments are conducted in an environment with specifications such as a 64-bit operating system, 16 GB RAM, and a 1.30 GHz Intel processor.

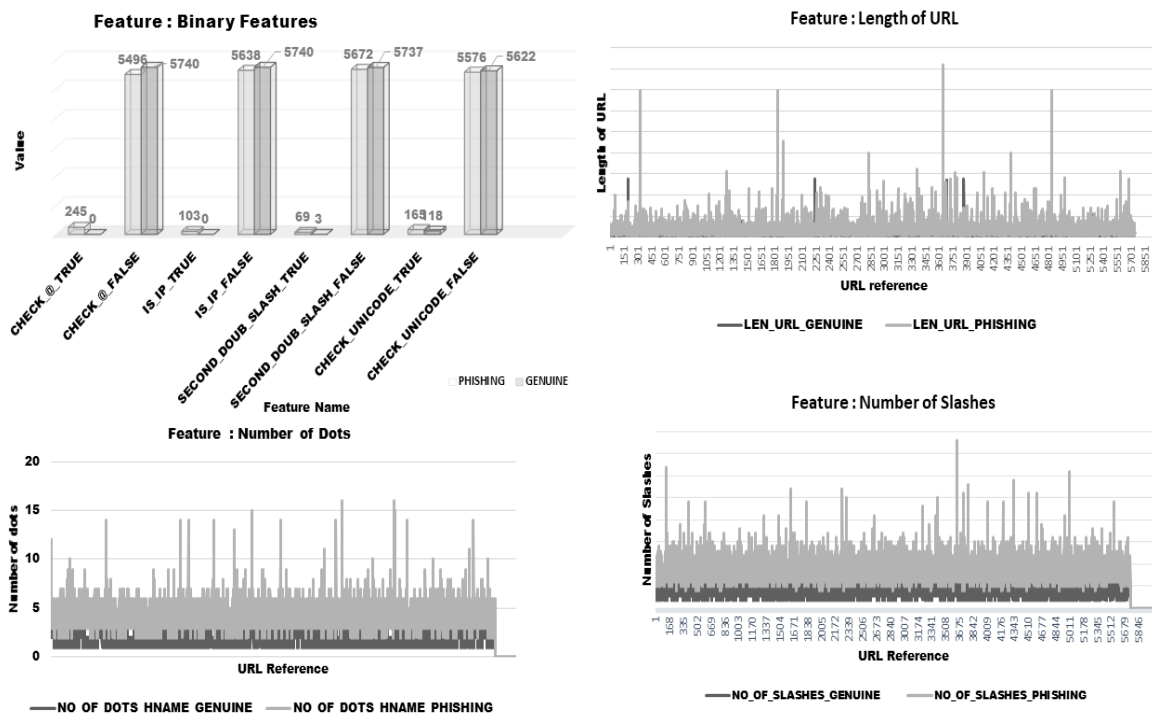
3.2. Feature identification and feature selection

The URLs collected from different sources are parsed, and the required features are retrieved. As the URLs are collected from different sources to preserve the unpredictability in the data set, we are not dependent on the feature data set and is generating our own feature set by combining different URL datasets and parsing the data. The seven features are finalized by analyzing the relative frequency of those in the dataset under consideration. The features selected included binary features as well as discrete value-based features. Table 1 summarizes the selected features and their relative presence in the previous studies.

The features selected are finalised by reinforcing their relative importance with other features. The evaluation is conducted on multiple standard datasets, and their relative importance is verified. Binary features like the presence of Unicode and a second double slash in URLs, as well as IP-based URLs, show a clear distinction between phishing and genuine URLs. Features like the length of the URL and the number of dots and slashes in the URL display a constant value range for genuine and phishing URLs to indicate the strength of the same in phishing URL detection. Figure 3 shows this feature analysis conducted on one training dataset. The same is repeated for the other datasets, too, to support the result. These handpicked features are finalized and forwarded to the next level to aid in phishing URL detection.

Table 1. Feature importance as confirmed by previous researchers

Feature Code	Feature Name	Description	References
IS_IP	IP based URL	Attackers replicate the page to lure users to avoid DNS server registration.	[10], [13], [17], [18], [20], [22]
LEN_URL	Length of URL	A tiny URL enhances suspicion, just as a very large URL.	[17], [20], [33]
CHECK_@	Presence of @ in URL	Browsers ignore any preceding character of '@' while parsing the URL, which helps the attacker to add a genuine-looking domain name before his malicious domain and dupe a victim.	[13], [17], [20], [22]
CHECK_UNICODE	Presence of Unicode characters in URL	Phishing domains tend to include Unicode to get a visual similarity to a genuine website.	[13], [17], [18], [20], [22]
NO_OF_DOTS_HNAME	Number of dots in host hostname	Including dots is a technique attackers adopt to hide the phishing domain inside a legitimate domain.	
SECOND_DOUB_SLASH	Presence of a second double slash in URL	Adding a second double slash in URL will confuse the crawlers with different versions.	[13], [18], [22], [33]
NO_OF_SLASHES	Number of slashes in URL	Number of slashes in a URL indicates the number of subdomains and is a direct indication that a URL is untrusted.	



TRAIN SET 1- FEATURE DISTRIBUTION

Figure 3. Feature distribution

3.3. Feature ranking

The identified feature probabilities are used for feature ranking. As the training data is not uniform, the result also shows heterogeneity in the probability values. The heterogeneous values and the reason for it are clearly visible from the feature summary statistics. The presence or absence of URL samples with individual features largely influences the probability values, as evident from the datasets. This replicates the real-world URL data, where the model will work, which has no predictability on the feature presence.

Logistic regression is applied to each data set separately, and attributes like coefficients, odds ratios, and probability values are calculated and analyzed. The calculated coefficient values, odds ratio, and probability are given in Table 2. The average value of probability is found to be representative and is used in the feature ranking phase. The feature probability tuple finalized is as given in Table 3.

Table 2. Coefficient, odds ratio and probabilities of training set

Data set	Attribute Names	IS_IP	LEN_U RL	CHECK_@	CHECK_ UNICODE	NO_OF_DOTS _HNAME	SECOND_ DOUB_SL ASH	NO_OF_ SLASHES
<u>TRAIN#1</u>	Coefficients	3.15	0.01	3.62	-0.39	0.18	1.97	0.17
<i>Phishing:</i> 5741	Odds Ratio	23.42	1.01	37.38	0.68	1.20	7.17	1.18
<i>Naïve:</i> 5740	Probabilities	0.96	0.50	0.97	0.40	0.55	0.88	0.54
<u>TRAIN#2</u>	Coefficients	0.00	0.00	0.89	0.57	0.00	0.53	0.04
<i>Phishing:</i> 55042	Odds Ratio	1.00	1.00	2.43	1.76	1.00	1.70	1.04
<i>Naïve:</i> 40868	Probabilities	0.50	0.50	0.71	0.64	0.50	0.63	0.51
<u>TRAIN#3</u>	Coefficients	-1.50	0.02	0.65	-0.07	0.27	0.08	1.51
<i>Phishing:</i> 980	Odds Ratio	0.22	1.02	1.92	0.93	1.31	1.08	4.52
<i>Naïve:</i> 858	Probabilities	0.18	0.51	0.66	0.48	0.57	0.52	0.82
<u>TRAIN#4</u>	Coefficients	0.95	0.02	0.14	0.18	-0.22	-1.14	0.16
<i>Phishing:</i> 3612	Odds Ratio	2.59	1.02	1.15	1.19	0.80	0.32	1.18
<i>Naïve:</i> 3297	Probabilities	0.72	0.50	0.53	0.54	0.45	0.24	0.54
<u>TRAIN#5</u>	Coefficients	3.01	0.00	2.66	-0.53	2.10	4.36	0.19
<i>Phishing:</i> 192830	Odds Ratio	20.28	1.00	14.27	0.59	8.17	78.44	1.20
<i>Naïve:</i> 179485	Probabilities	0.95	0.50	0.93	0.37	0.89	0.99	0.55

Table 3. Selected features probability values

Feature_Name	Probability value (Average)
CHECK_@	0.7616
IS_IP	0.6632
SECOND_DOUB_SLASH	0.6514
NO_OF_SLASHES	0.5916
NO_OF_DOTS_HNAME	0.5900
LEN_URL	0.5028
CHECK_UNICODE	0.4878

3.4. PUP-phishing URL prediction

The MS of the URL is calculated and is alerted if it is more than the accepted threshold value. The threshold value for MS is calculated by feeding the testing set URLs to the finite state machine (PHISH_FSA) created and the threshold values are finalized. The datasets TEST#1 and TEST#2 returned a threshold value of 2, meaning any URL evaluation results in a MS greater than 2 is suspected as malicious URLs. Figure 4 (see in *Appendix*) represents the calculated MS value for the different URL datasets under consideration, with url reference number in the X axis and MS on the Y axis. The malicious score value distribution for the testing datasets. The result confirms a threshold value of 2 is enough for a URL to be categorized as a phishing URL.

4. CONCLUSION

As this model performs the detection based on the self-generated feature set, this model shows different performance indicators compared to the parallel research findings. Phishing techniques are evolving daily and attackers are finding new ways to obfuscate the irregularities in the URL. This model put forth a highly adaptable model for these changes which can accommodate the new features coming up and provide promising results. The change adaptability is guaranteed by continuous checking and revising of the feature weights and threshold values.

The naive idea of real time phishing URL detection using finite state automata is implemented successfully in this model. The real time analysis of the URL gives the advantage to the model as the model will not be biased towards a single data set used in the training phase. The model experimentation shows promising rates of false positives while tested with the naive data set. The false negatives still need to be improved and reason found to be the versatility of the phishing URLs we collect and evaluate. However, the PHISH_FSA is modelled so that these adjustments can be easily accommodated, and the model can be tuned.

The FIR phase is also developed, considering that these new features should be accommodated without many changes in the model.

The model can be implemented to find the MS of the URL, and the administrator can decide the threshold and either accept or reject any new URL entering the organizational network territory. The model needs to be constantly tuned with new datasets to include new features that the attackers can try, and it also needs to revamp the PHISH_FSA at regular intervals so that the error rates are reduced.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Nisha T N	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓				
Dhanya Pramod	✓	✓			✓					✓	✓	✓	✓	✓

C : Conceptualization

M : Methodology

So : Software

Va : Validation

Fo : Formal analysis

I : Investigation

R : Resources

D : Data Curation

O : Writing - Original Draft

E : Writing - Review & Editing

Vi : Visualization

Su : Supervision

P : Project administration

Fu : Funding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

Data availability is not applicable to this paper as no new data were created or analyzed in this study.

REFERENCES

- [1] Bakerhostetler, "BakerHostetler's 2025 data security incident response report finds less malware used in 2024", 2025, [Online]. Available: <https://www.bakerlaw.com/insights/bakerhostetler-launches-2024-data-security-incident-response-report-persistent-threats-new-challenges/>. Accessed: Nov. 10, 2024.
- [2] "Enterprise security solutions," Ibm.com. [Online]. Available: <https://www.ibm.com/security>. Accessed: Nov. 13, 2024.
- [3] F. L. Greitzer, J. R. Strozer, S. Cohen, A. P. Moore, D. Mundie, and J. Cowley, "Analysis of unintentional insider threats deriving from social engineering exploits," in *2014 IEEE Security and Privacy Workshops*, 2014.
- [4] "APWG," Apwg.org. [Online]. Available: <https://apwg.org/trendsreports>. Accessed: Oct. 13, 2024.
- [5] O. Christou, N. Pitropakis, P. Papadopoulos, S. McKeown, and W. Buchanan, "Phishing URL detection through top-level domain analysis: A descriptive approach," in *Proceedings of the 6th International Conference on Information Systems Security and Privacy*, 2020, doi: 10.5220/0008902202890298.
- [6] A. Yasin and A. Abuhasan, "An intelligent classification model for phishing email detection," *Int. J. Netw. Secur. Appl.*, vol. 8, no. 4, pp. 55–72, 2016, doi: 10.5121/ijnsa.2016.8405.
- [7] S. Atawneh and H. Aljehani, "Phishing email detection model using deep learning," *Electronics (Basel)*, vol. 12, no. 20, p. 4261, 2023, doi: 10.3390/electronics12204261.
- [8] Y. Fang, C. Zhang, C. Huang, L. Liu, and Y. Yang, "Phishing email detection using improved RCNN model with multilevel vectors and attention mechanism," *IEEE Access*, vol. 7, pp. 56329–56340, 2019, doi: 10.1109/ACCESS.2019.2913705.
- [9] N. B. Harikrishnan, R. Vinayakumar, and K. P. Soman, "A machine learning approach towards phishing email detection," *Proceedings of the Anti-Phishing Pilot at ACM International Workshop on Security and Privacy Analytics (IWSPA AP)*, vol. 2013, pp. 455–468, 2018.
- [10] J. Lee, F. Tang, P. Ye, F. Abbasi, P. Hay, and D. M. Divakaran, "D-Fence: A Flexible, efficient, and comprehensive phishing email detection system," in *2021 IEEE European Symposium on Security and Privacy (EuroS&P)*, IEEE, 2021, pp. 578–597.
- [11] D. L. Cook, V. K. Gurbani, and M. Daniluk, "Phishwish: A stateless phishing filter using minimal rules," in *Financial Cryptography and Data Security*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 182–186.
- [12] P. Agrawal and D. Mangal, "A novel approach for phishing URLs Detection," *International Journal of Science and Research (IJSR)*, vol. 5, no. 5, pp. 1117–1122, 2015.
- [13] M. S. Kumar and B. Indrani, "Frequent rule reduction for phishing URL classification using fuzzy deep neural network model," *Iran J. Comput. Sci.*, vol. 4, no. 2, pp. 85–93, 2021, doi: 10.1007/s42044-020-00067-x.
- [14] A. Karim, M. Shahroz, K. Mustofa, S. B. Belhaouari, and S. R. K. Joga, "Phishing detection system through hybrid machine learning based on URL," *IEEE Access*, vol. 11, pp. 36805–36822, 2023, doi: 10.1109/ACCESS.2023.3252366.

- [15] S. Jalil, M. Usman, and A. Fong, "Highly accurate phishing URL detection based on machine learning," *J. Ambient Intell. Humaniz. Comput.*, vol. 14, pp. 9233-9215, 2023, doi: 10.1007/s12652-022-04426-3.
- [16] L. Xu, Z. Zhan, S. Xu, and K. Ye, "Cross-layer detection of malicious websites," in *Proceedings of the third ACM conference on Data and application security and privacy*, 2013, doi: 10.1145/2435349.2435366.
- [17] R. M. Mohammad, F. Thabtah, and L. McCluskey, "Intelligent rule-based phishing websites classification," *IET Inf. Secur.*, vol. 8, no. 3, pp. 153-160, 2014, doi: 10.1049/iet-ifs.2013.0202.
- [18] P. Prakash, M. Kumar, R. R. Kompella, and M. Gupta, "PhishNet: Predictive blacklisting to detect phishing attacks," in *2010 Proceedings IEEE INFOCOM*, 2010, pp. 1-5, doi: 10.1109/INFOCOM.2010.5462216.
- [19] M. Aburrous, M. A. Hossain, K. Dahal, and F. Thabtah, "Intelligent phishing detection system for e-banking using fuzzy data mining," *Expert Syst. Appl.*, vol. 37, no. 12, pp. 7913-7921, 2010, doi: 10.1016/j.eswa.2010.04.044.
- [20] J. Hong, T. Kim, J. Liu, N. Park, and S. W. Kim, "Phishing url detection with lexical features and blacklisted domains". *Adaptive autonomous secure cyber systems*, pp. 253-267, 2020, doi: 10.1007/978-3-030-33432-1_12.
- [21] G. Xiang, J. Hong, C. P. Rose, and L. Cranor, "Cantina+ a feature-rich machine learning framework for detecting phishing web sites," *ACM Transactions on Information and System Security (TISSEC)*, vol. 14, no. 2, pp. 1-28, 2011, doi: 10.1145/2019599.2019606.
- [22] A. K. Jain, S. Parashar, P. Katore, and I. Sharma, "PhishSKaPe: A content-based approach to escape phishing attacks," *Procedia Comput. Sci.*, vol. 171, pp. 1102-1109, 2020, doi: 10.1016/j.procs.2020.04.118.
- [23] Y. Zhang, J. I. Hong, and L. F. Cranor, "Cantina: a content-based approach to detecting phishing web sites," in *Proceedings of the 16th international conference on World Wide Web*, 2007, pp. 639-648, doi: 10.1145/1242572.1242659.
- [24] B. Wardman, T. Stallings, G. Warner, and A. Skjellum, "High-performance content-based phishing attack detection," in *2011 eCrime Researchers Summit*, 2011, doi: 10.1109/eCrime.2011.6151977.
- [25] Y. Cao, W. Han, and Y. Le, "Anti-phishing based on automated individual white-list," in *Proceedings of the 4th ACM workshop on Digital identity management*, 2008, pp. 51-60, doi: 10.1145/1456424.1456434.
- [26] G. Xiang and J. I. Hong, "A hybrid phish detection approach by identity discovery and keywords retrieval," in *Proceedings of the 18th International Conference on World Wide Web*, 2009, doi: 10.1145/1526709.152678.
- [27] S. Afroz and R. Greenstadt, "PhishZoo: Detecting phishing websites by looking at them," in *2011 IEEE Fifth International Conference on Semantic Computing*, 2011, pp. 368-375, doi: 10.1109/ICSC.2011.52.
- [28] M. Dunlop, S. Groat, and D. Shelly, "GoldPhish: Using images for content-based phishing analysis," in *2010 Fifth International Conference on Internet Monitoring and Protection*, 2010.
- [29] A. Ozcan, C. Catal, E. Donmez, and B. Senturk, "A hybrid DNN-LSTM model for detecting phishing URLs," *Neural Comput. Appl.*, vol. 35, no. 7, pp. 4957-4973, 2023, doi: 10.1007/s00521-021-06401-z.
- [30] H. Le, Q. Pham, D. Sahoo, and S. C. H. Hoi, "URLNet: Learning a URL representation with deep learning for malicious URL Detection," arXiv [cs.CR], 2018.
- [31] F. Tajaddodianfar, J. W. Stokes, and A. Gururajan, "Texception: A character/word-level deep learning model for phishing URL detection," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 2857-2861, doi: 10.1109/ICASSP40776.2020.9053670.
- [32] H. Shahriar and M. Zulkernine, "PhishTester: Automatic testing of phishing attacks," in *2010 Fourth International Conference on Secure Software Integration and Reliability Improvement*, 2010, pp. 198-207, doi: 10.1109/SSIRI.2010.17.
- [33] Kaggle.com. [Online]. Available: <https://www.kaggle.com/>. Accessed: Sep. 13, 2024.
- [34] Phishtank.com. [Online]. Available: <https://www.phishtank.com/>. Accessed: Oct. 13, 2024.
- [35] "Common crawl - overview," Commoncrawl.org. [Online]. Available: <https://commoncrawl.org/the-data/>. Accessed: May. 13, 2024.

APPENDIX

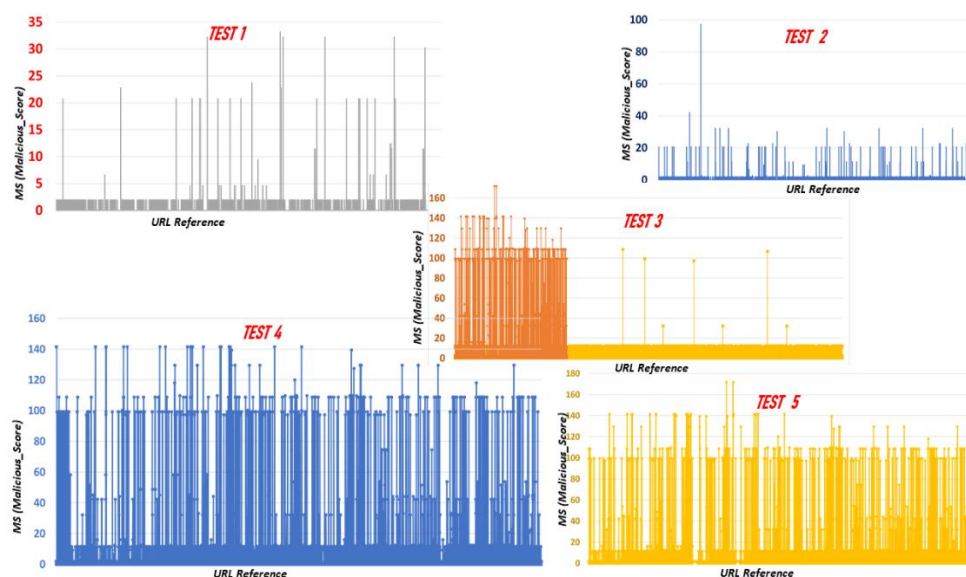








Figure 4. Malicious score distribution

BIOGRAPHIES OF AUTHORS

Nisha T N    she works as Assistant Professor at Symbiosis Centre for Information Technology (SCIT), a constituent of the Symbiosis International University (SIU), Pune. She completed a Ph.D. in Computer Science from Symbiosis International University in network intrusion detection. She has a teaching experience of fifteen years in the areas such as information security, ethical hacking, programming concepts, optimization and cyber intelligence. She can be contacted at email: nisha@scit.edu.



Dhanya Pramod    she is a Professor and Director at the Symbiosis Centre for Information Technology (SCIT), a constituent of the Symbiosis International University (SIU), Pune. She has a Ph.D. in Computer Science from Symbiosis International University, India and her teaching and research interests are information security, networks and application security and predictive analytics. She can be contacted at email: dhanyaspramod@gmail.com.