Malicious Website Detection Based on URL Classification: A Comparative Analysis

Swati Maurya and Anurag Jain

Abstract Phishing has been one of the most frequent cyber threats in the recent decade, prompting an increase in anti-phishing research and the development of numerous solutions for detecting and preventing phishing assaults. This paper identifies the system's vulnerabilities and adversaries' tactics to deceive Internet users into trusting the malicious email or website and providing sensitive information and credentials. For this study, the relevant URL features are retrieved from the collected dataset that includes phishing and legitimate URLs of websites. The correlation among different features is studied that can help users to identify fake web URLs by scanning phishing specific properties. This paper also analyzes the performance outcome of the machine learning, ensemble, and deep learning techniques on the collected dataset. Each model's performance is compared and measured, and random forest and gradient boosting with XGBoost are found to be the best optimal model for phishing binary classification problem in terms of accuracy (97.3%).

Keywords Anti-phishing · Phishing detection · Machine learning · Deep learning · URL-based classification

1 Introduction

Phishing is a technique adversaries used to gain personal and financial information such as login credentials and payment card details by impersonating the user or by tricking them into trusting fake websites or emails. It is a social engineering act in which an attacker uses a specially crafted message to random people in the hopes of obtaining sensitive information or utilizing the vulnerability of the user system for deploying and executing malicious software on the victim's infrastructure, such as

S. Maurya (✉) · A. Jain
Guru Gobind Singh Indraprastha University, New Delhi, New Delhi, India
e-mail: swatimaurya@hotmail.com

A. Jain
e-mail: anurag@ipu.ac.in

P. K. Singh et al. (eds.), *Proceedings of Third International Conference on Computing, Communications, and Cyber-Security*, Lecture Notes in Networks and Systems 421,
https://doi.org/10.1007/978-981-19-1142-2_19

viruses, Trojans, and ransomware [1]. Fake emails have increased where the scammer pretends to be a reliable and legitimate government or bank official, and facts are added to support their claim [2]. The scammer may notify the user that their account has been used illegally or questionably. The email contains details to convince the user that a significant purchase was made in another region and ask if the user has approved the payment. The fraudsters are prompt to confirm the credit card or bank account information so the 'bank' can examine the whole scenario. This way, they steal sensitive information from users.

Every day, cybercriminals conduct thousands of phishing campaigns like this, and most of them are successful. Attackers often modify their techniques, but specific characteristics might help the user identify a malicious email, text message, or website. User education to follow best practices while browsing the Internet and following safety guidelines can assist users to avoid getting trapped in attempts to steal sensitive information [3]. Advanced attack methods and assault tactics, which cannot be diagnosed by primary education and training, necessitate automated detection and prevention approaches. Software-based defense mechanisms help in detecting malicious emails, fake text messages, and phishing websites.

Phishing Tactics: A typical phishing assault might use various tactics, such as exploiting browser vulnerabilities or executing man-in-the-middle attacks. However, the most basic and often used technique is to create a webpage that looks identical to the one that the user is familiar with or craft emails or text messages in a way that appears to be genuine and helps them gain the trust of the user [4]. They hide URLs of fake external websites which replicate the appearance and user interface of the original website to trick the user into entering their details and credentials. The most popular phishing attacks are shown in Fig. 1.

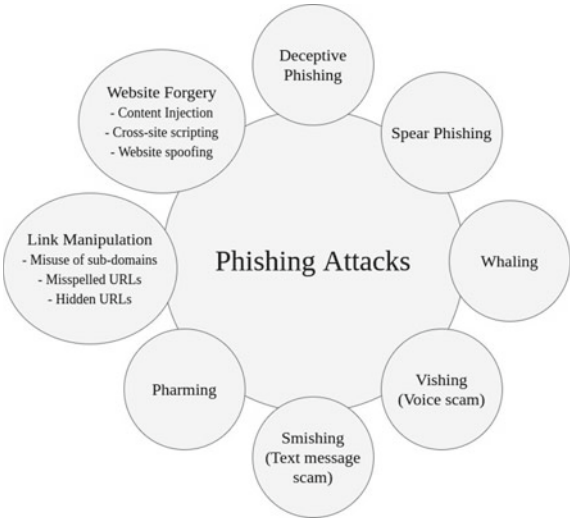


Fig. 1 Common phishing attacks used by adversaries for stealing sensitive information from the users

Phishing remains a severe security issue, and many Internet users are still victims of this deception. Furthermore, such attacks cause severe problems for Internet users and organizations that offer financial services over the Internet.

Filter-based phishing detection [5] techniques are incorporated these days into most email service providers to transfer suspicious emails to a 'Junk' or 'Spam' folder. When email filtering is enabled, incoming emails are scanned independently for features that indicate malicious content and transfer those emails to a different folder. Browser extension [3] and webpage content analysis [6]-based phishing detection and prevention solutions provide security from phishing websites if they match the suspicious criteria. Despite numerous anti-phishing solutions available these days for prevention from phishing attacks, the adversaries always stay a step ahead which makes all anti-phishing solutions incapable of preventing zero-day attacks. Hence, the need to deeply understand the correlation among different features present in a phishing webpage arises. The appropriate models should be selected while drafting a new adaptable anti-phishing solution making it more suitable for zero-day attack prevention.

This paper will analyze the effectiveness of phishing detection techniques based on URL classifications. With the advancements in machine learning algorithms and their accurate predictions in less time, the phishing domain research in the last decade has shifted to the machine learning, ensemble, and deep learning-based solutions. Section 2 presents the research methodology followed during this study and analysis. Section 3 discusses the details for feature analysis from URLs and the correlation among them. Section 4 covers popular classification models for phishing detection, and their performance is comparatively analyzed when used on the same phishing URL dataset. It analyzes the performance metrics and evaluates the results obtained for classification models on the URL dataset to find the best classifier. Section 5 summarizes the paper.

2 Research Methodology

Various research studies have been conducted by authors earlier to analyze the phishing detection techniques, and the approaches followed are as follows. As discussed in [7], the authors compared software-based phishing detection techniques like blacklists, heuristic detection techniques, visual similarity detection techniques, and data mining detection techniques available in the literature. A detailed survey of literature available for phishing detection approaches is also presented in [8]. Most of the surveys or analyses are focused on techniques mentioned in the literature, but there is no comparative analysis available that should guide toward selecting the best optimal model while designing an anti-phishing solution. This paper's study will aid academics and industry in determining the optimum algorithm that can be used for anti-phishing solutions based on requirements and resources.

This section briefs the steps followed in this study for URL feature analysis and performance comparison of latest machine learning, ensemble, and deep learning-

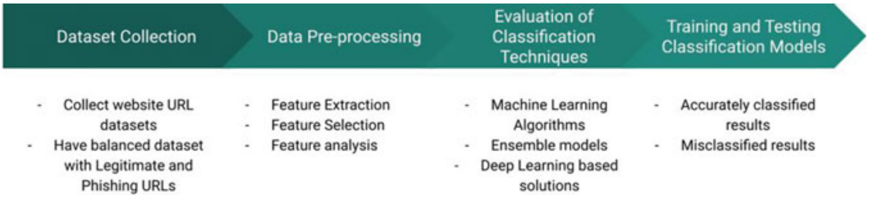

Dataset Collection
Data Pre-processing
Evaluation of Classification Techniques
Training and Testing Classification Models
- Collect website URL datasets
- Have balanced dataset with Legitimate and Phishing URLs
- Feature Extraction
- Feature Selection
- Feature analysis
- Machine Learning Algorithms
- Ensemble models
- Deep Learning based solutions
- Accurately classified results
- Misclassified results

Fig. 2 Research methodology followed in the paper for URL feature analysis and performance comparison of latest classification models on phishing dataset

based classification models on phishing dataset provided by PhishTank [9], dmoztools.net [10]. Figure 2 presents the sequence of steps followed, and detailed explanation is given in following sections.

3 Analysis of Features Extracted from Web URLs

The URL and webpage content for fake websites are usually replicated to appear similar to the original website [11]. This research focuses on phishing website detection in real time by examining the features of the URL of the webpage. The malicious websites can be efficiently detected by thoroughly analyzing their URL. The attackers cannot utilize the exact URL of an original site, and they frequently misspell URL elements such as ‘PrimaryDomain,’ ‘SubDomain,’ and ‘PathDomain’ [12]. Identifying these phishing URL alteration tactics will undoubtedly assist in educating individuals and organizations about phishing attacks and ensuring prevention from them.

Features are extracted from the collected dataset of URLs and classified based on categories defined in [3] and are shown in Fig. 3. The analysis is done to understand the importance of each feature in classification for phishing or legitimate URL and the correlation among different features. Lexical characteristics of the URL [13] are analyzed in this study, and the efficacy for phishing prediction is studied. The observations from examining features of web URLs in the collected dataset are:

- Each data sample contains 30 features and a class label ‘result’ that indicates whether or not it is a phishing website (1 or -1).
- Size of URL: Long URLs are frequently associated with concealing the suspicious section of a fake website URL in the address bar to mislead the user.
- Number of dots: In comparison with legitimate websites, phishing pages frequently have more than 5–6 dots in their URLs.
- IP in the domain: Using an IP address instead of a domain name in a URL indicates an effort to steal personal information.

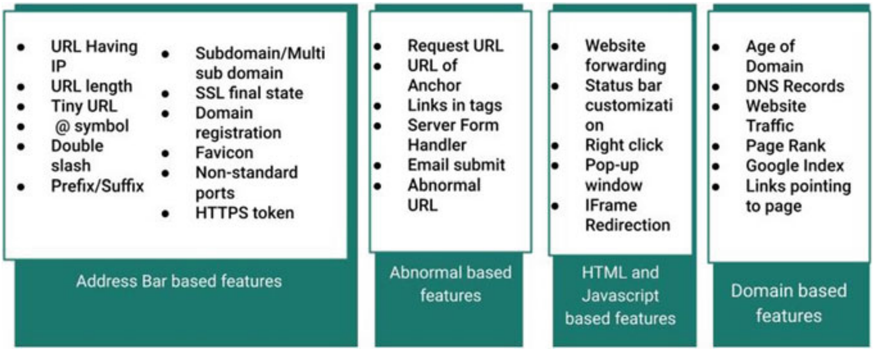


Fig. 3 Classification of website features based on address bar, abnormal, HTML and JavaScript and domain features

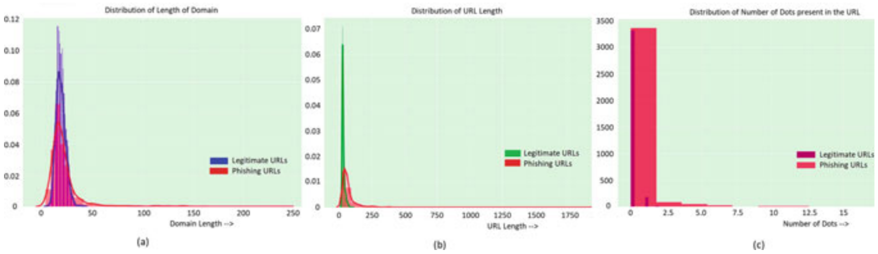


Fig. 4 Visual representation of distribution of **a** length of domain, URL length (**b**), and **c** number of dots present in legitimate and phishing URLs in the collected dataset

- Special characters in URL: The phishing link will perplex the user by adding a special character in the URL. "@" hides the phishing URL by commenting out the domain name that comes before it. The presence of the 'hyphen' and '@' symbol in the URL dominate in malicious URLs, whereas legitimate URLs avoid using them [12].
- Double slash ('//'): The presence of a double slash in a URL route indicates that the visitor will be redirected to a different website.
- Multiple sub-domains and a domain name mismatch: Phishers employ this type of technique to persuade victims that the message or email they received originated from a well-known organization. They use the genuine organization's domain name and append multiple sub-domains as a prefix to deceive users into thinking the crafted fake URL is genuine.
- Age of a URL: Phishing websites have been found to only exist for a short time, whereas trustworthy websites are registered and paid for several years in advance.

The distribution of domain length, URL length, and the number of dots present in legitimate and phishing web URLs is shown in Fig. 4a–c, respectively.

Correlation between URL features: The statistical measure of a linear relationship between two variables is known as correlation, and the visual representation is called

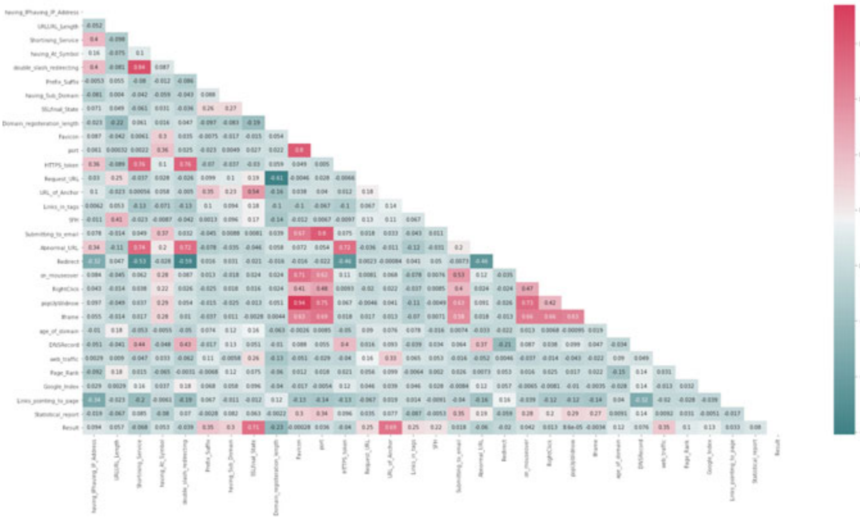


Fig. 5 Correlation heatmap representing the interdependence between URL features of the collected dataset

correlation heatmap [14]. It is the measure of interdependence between two variables. The matrix data format is utilized when there are numerous variables. Figure 5 shows a custom diverging colormap in the form of a matrix generated for the collected dataset and is drawn with the mask and correct aspect ratio. The correlation coefficient might have any value between −1 and 1 [15].

- Value = 1: The correlation between two variables is considered to be positive and indicates that while one variable rises, the other increases as well.
- Value = −1: Negative correlation between two variables is defined as a value of −1 and indicates that as one variable goes up, the other goes down.
- Value = 0: There is no connection between two variables if the value is 0 and indicates that the variables vary at random in relation to one another.

Identifying the characteristics of phishing URL alteration methods and their correlations can aid users in recognizing phishing attempts just by looking at them.

4 Classification Models for Phishing Detection Based on Web URLs

Phishing is a binary classification problem that classifies the given sample into two classes: legitimate or phishing. This research is focused on analyzing the performance of the latest classification algorithms provided by machine learning, ensemble techniques, and deep learning models that are best suited for binary classification.

Machine Learning (ML) Techniques: Data analysis is made easier and more efficient using machine learning. The capacity to construct adaptable models for specific tasks like phishing detection is a fundamental feature of machine learning. ML models might swiftly adapt to changes to identify patterns, which would aid in developing a learning-based identification system [3]. The following algorithms have been chosen because of their accurate prediction results for binary problems: decision tree, random forest, K-nearest neighbor, logistic regression, support vector machine.

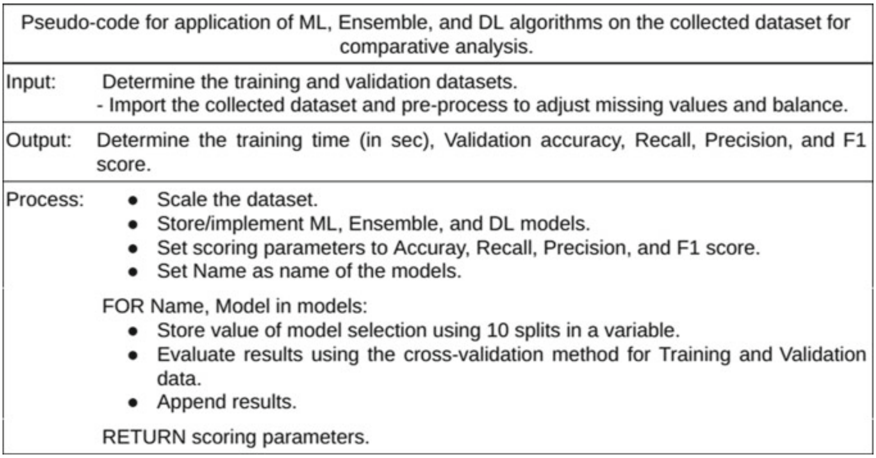
Ensemble Classification Techniques: An ensemble is made up of several hypotheses that are created from training data using a primary learning method. Most ensemble methods generate homogeneous ensembles using a single base learning algorithm; however, other approaches employ several learning algorithms to produce heterogeneous ensembles. Several high-performance and advanced frameworks like AdaBoost, XGBoost, and a family of gradient boosting techniques [16] that focus on both speed and accuracy have recently been analyzed on the collected dataset for their performance.

Deep Learning (DL) Techniques: Artificial neural networks are used in deep learning models to conduct complex computations on large datasets [17]. It is a form of ML based on the human brain's structure and function. Algorithms extract features, organize objects, and find valuable data patterns during the training phase by using unknown elements in the input distribution. Machines are trained using examples. DL algorithms require high-end infrastructure to train in an acceptable amount of time. When there is a dearth of domain expertise for feature introspection, DL approaches shine since feature engineering is less of a concern [18].

4.1 Performance Comparison of Classification Models

The malicious and legitimate URLs are collected from PhishTank and dmoztools.net, and a dataset is formed. The collected dataset consists of a total of 65428 web URLs which have 29182 phishing URLs and 36246 legitimate URLs. The dataset is preprocessed, and features are extracted and chosen for analysis. The collected features from URLs are concatenated after random shuffling during the feature extraction step to prevent the overfitting [19] problem during model training. This also helps balance the distribution while splitting the data into training (75%) and testing (25%) sets. The implementation is done in Python with the help of machine learning libraries, and the pseudo-code is presented in Fig. 6.

Table 1 gives a comparison summary of performance metrics obtained as a result of applying classification algorithms for ML, ensemble, and DL techniques on the training set and validation set. Figure 7 shows visual representation of accuracies obtained from machine learning, ensemble, and deep learning techniques applied on collected dataset.



Pseudo-code for application of ML, Ensemble, and DL algorithms on the collected dataset for comparative analysis.

```
Input:    Determine the training and validation datasets.
          - Import the collected dataset and pre-process to adjust missing values and balance.
Output:   Determine the training time (in sec), Validation accuracy, Recall, Precision, and F1
          score.
Process:     •  Scale the dataset.
             •  Store/implement ML, Ensemble, and DL models.
             •  Set scoring parameters to Accuray, Recall, Precision, and F1 score.
             •  Set Name as name of the models.

          FOR Name, Model in models:
             •  Store value of model selection using 10 splits in a variable.
             •  Evaluate results using the cross-validation method for Training and Validation
                data.
             •  Append results.

          RETURN scoring parameters.
```

Fig. 6 Pseudo-code used in experiments on the collected dataset for comparative analysis

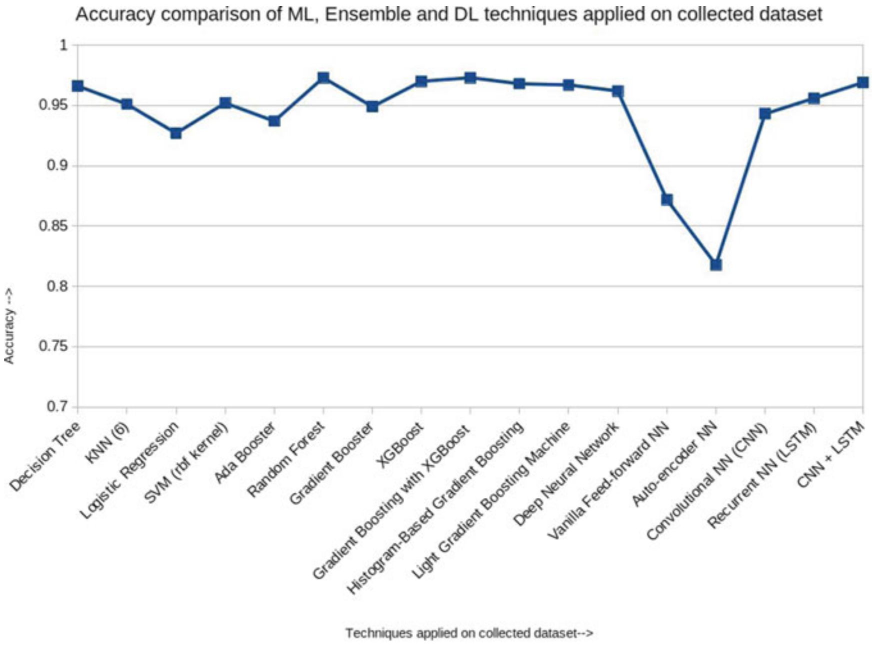


Fig. 7 Visual representation of accuracies obtained from machine learning, ensemble, and deep learning techniques applied on collected dataset

Table 1 Performance comparison of machine learning, ensemble, and deep learning techniques applied on collected dataset

Classifier	Training time (s)	Validation accuracy	Recall	Precision	$F1$ score
Machine learning classifiers					
Decision tree	0.017	0.966	0.971	0.968	0.969
KNN (5)	0.021	0.951	0.959	0.953	0.956
Logistic regression	0.095	0.927	0.944	0.926	0.935
SVM (rbf kernel)	1.509	0.952	0.969	0.947	0.957
Ensemble classifiers					
Ada booster	0.266	0.937	0.955	0.933	0.943
Random forest	0.399	0.973	0.981	0.967	0.974
Gradient booster	0.784	0.949	0.963	0.947	0.955
XGBoost	0.418	0.97	0.976	0.968	0.971
Gradient boosting with XGBoost	0.412	0.973	0.982	0.969	0.974
Histogram-based gradient boosting	0.396	0.968	0.975	0.967	0.971
Light gradient boosting machine	0.117	0.967	0.975	0.966	0.971
Deep learning classifiers					
Deep neural network	10.220	0.962	0.95	0.972	0.969
Vanilla feed-forward NN	12.54	0.872	0.868	0.869	0.867
Auto-encoder NN	8.657	0.818	0.825	0.802	0.821
Convolutional NN (CNN)	5.984	0.943	0.951	0.945	0.949
Recurrent NN (LSTM)	7.217	0.956	0.969	0.951	0.96
CNN + LSTM	5.844	0.969	0.979	0.965	0.972

4.2 Discussion and Results Analysis

This research evaluated the time taken for training different classification models (in seconds), validation accuracy obtained, recall, precision, and $F1$ score. Table 1 lists the results of the experiments mentioned above, and the following are the observations.

ML models get quickly trained, allowing them to make predictions and self-improve algorithms.

- Compared to other ML algorithms, the decision tree classifier (C4.5) predicts the phishing website accurately and fastest in training. It uses the 'Gini measure of impurity,' which helps the tree create 'pure nodes' with only one class label that does not need to be further divided, hence the fast execution [3].

- SVM has been tested with different kernels, but RBF kernel gave the best results compared to linear, poly, and sigmoid kernels.
- For testing KNN, there is no ideal number for setting k suitable for all types of datasets. Various experiments were conducted by altering the value of k to find the best-suited value for the collected dataset. A perfect balance needs to be found as noise has a greater influence on the outcome when the number of neighbors is small; moreover, a large number of neighbors makes obtaining the result computationally expensive [20].

Ensemble Models show better results as compared to standard ML models. Random forest (RF) and gradient boosting with XGBoost are the best in terms of accuracy (97.3%) and even take almost the same time for getting trained.

- RF adds randomness to the training and validation dataset and uses more trees, reducing variance, ultimately making the predictions fast and noise prune.
- The significant benefit of XGBoost over other algorithms is its rapid speed, as well as the 'regularization parameter,' which successfully lowers 'variance.' Usage of learning rate and subsamples from features like RF allows it to generalize even further. Hyperparameter tuning increases performance, and hybrid with gradient boosting algorithms reduces the training time and results in higher accurate predictions.

DL models take maximum time to get train due to the large number of hyperparameters. They incrementally learn high-level characteristics from data through the hidden layer architecture. The performance of DL models has been observed to increase with the amount of data [18].

- CNN combined with LSTM gave the best prediction results as compared to separately testing CNN or LSTM. The grid pattern analysis of CNN, when combined with feedback connections of LSTM, takes less time to train, and the performance increases.
- Although DL models take more time to train, once trained, their accuracy keeps on increasing with time as they can learn from past observations and utilize them for future predictions.
- DL models work best with a large amount of data, and the dataset used in this study was not that huge. With small datasets, the results are less accurate than ensemble models.

This study can be utilized before finalizing any model before drafting any anti-phishing solution based on URL characteristics. The comparative analysis highlights the best classifier in each section which can be chosen based on requirements and resources for the research. These results were retrieved using classifiers on a live dataset; thus, they are not theoretical, which enhances the study's reliability and dependability.

5 Conclusion

Recently with an increased emergence of phishing attacks through emails, fake websites, text messages, and phone calls, the need for awareness among Internet users has risen to identify a fake email or webpage. This research covered the basic phishing attack scenarios that deceive users into trusting scammers or adversaries. The URL feature analysis presents the correlation between web URL features which help any user scan the links provided in email before clicking them. A thorough understanding of these interdependencies among features prevents users from falling prey to fake websites that look similar to the original webpages but steal sensitive information. This study also analyzed the performance of the latest ML, ensemble, and DL algorithms. Their speed and accuracy are compared for the same dataset. Random forest and gradient boosting with XGBoost are found to be the best optimal solution for classifying URL-based phishing detection in terms of accuracy (97.3%). The limitation of this research is that comparative analysis was limited only to feature analysis of the web URLs. The classification techniques were tested on collected web URL datasets which can be extended to cover website content in the future.

References

1. Khan, F., Ncube, C., Ramasamy, L. K., Kadry, S., & Nam, Y. (2020). A digital DNA sequencing engine for ransomware detection using machine learning. *IEEE Access, 8,* 119710–119719.
2. Hoang, L. N., Faucon, L., Jungo, A., Volodin, S., Papuc, D., Liossatos, O., Crulis, B., Tighanimine, M., Constantin, I., Kucherenko, A., & Maurer, A. (2021). Tournesol: A quest for a large, secure and trustworthy database of reliable human judgments. arXiv preprint arXiv:2107.07334
3. Maurya, S., Singh, H., & Jain, A. (2019). Browser extension based hybrid anti-phishing framework using feature selection. *International Journal of Advanced Computer Science and Applications, 10*(11).
4. 12 Types of Phishing Attacks to watch out for. https://www.helixstorm.com/blog/x-types-of-phishing-attacks-to-watch-out-for/ Last Accessed August 9, 2021.
5. Surwade, A. U. (2020). Blocking Phishing e-mail by extracting header information of e-mails. In *2020 International Conference on Smart Innovations in Design, Environment, Management, Planning and Computing* (pp. 151–155).
6. Opara, C., Wei, B., & Chen, Y. (2020). HTMLPhish: Enabling phishing web page detection by applying deep learning techniques on HTML analysis. In *2020 International Joint Conference on Neural Networks* (pp. 1–8).
7. Khonji, M., Iraqi, Y., & Jones, A. (2013). Phishing detection: a literature survey. *IEEE Communications Surveys & Tutorials, 15*(4), 2091–2121.
8. Vijayalakshmi, M., Shalinie, S. M., & Yang, M. H. (2020). Web phishing detection techniques: A survey on the state-of-the-art, taxonomy and future directions. *IET Networks, 9*(5), 235–246.
9. PhishTank, https://phishtank.org/. Last accessed July 6, 2021.
10. DMOZ, https://dmoztools.net/. Last accessed July 20, 2021.
11. Tomaselli, J., Willoughby, A., Amezcua, J. V., Delehanty, E., Floyd, K., Wright, D., Lammers, M., & Vetter, R. (2021). Verifying phishmon: A framework for dynamic webpage classification. In *Proceedings of the 2021 ACM Southeast Conference* (pp. 185–189).
12. Mohammad, R. M., Thabtah, F., & McCluskey, L. (2015). *Phishing websites features*. School of Computing and Engineering, University of Huddersfield.

13. Khonji, M., Iraqi, Y., & Jones, A. (2011). Lexical URL analysis for discriminating phishing and legitimate websites. In *Proceedings of the 8th Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference* (pp. 109–115).
14. Haarman, B. C. B., Riemersma-Van der Lek, R. F., Nolen, W. A., Mendes, R., Drexhage, H. A., & Burger, H. (2015). Feature-expression heat maps—A new visual method to explore complex associations between two variable sets. *Journal of Biomedical Informatics, 53*, 156–161.
15. Kumar, A. (2021). *Correlation concepts, matrix and heatmap using seaborn*. https://vitalflux.com/correlation-heatmap-with-seaborn-pandas/. Last accessed August 1, 2021.
16. Bentéjac, C., Csörgő, A., & Martínez-Muñoz, G. (2021). A comparative analysis of gradient boosting algorithms. *Artificial Intelligence Review, 54,* 1937–1967.
17. Maurya, S., & Jain, A. (2020). Deep learning to combat phishing. *Journal of Statistics and Management Systems, 23*(6), 945–957.
18. Mahapatra, S. (2021). *Why deep learning over traditional machine learning?*https://towardsdatascience.com/why-deep-learning-is-needed-over-traditional-machine-learning-1b6a99177063. Last accessed June 3, 2021.
19. Yeom, S., Giacomelli, I., Menaged, A., Fredrikson, M., & Jha, S. (2020). Overfitting, robustness, and malicious algorithms: A study of potential causes of privacy risk in machine learning. *Journal of Computer Security, 28*(1), 35–70.
20. Shahrivari, V., Darabi, M. M., & Izadi, M. (2020). *Phishing detection using machine learning techniques*. arXiv preprint arXiv:2009.11116