Determining the Most Effective Machine Learning Techniques for Detecting Phishing Websites

S. M. Mahamudul Hasan, Nirjas Mohammad Jakilim, Md. Forhad Rabbi, and Rumel M. S. Rahman Pir

1 Introduction

Phishing is one of the most serious cybersecurity threats since it includes creating fake websites that seem to be legitimate [1]. In this assault, the user inputs important information such as credit card numbers, passwords, and so on to a bogus website that seems to be real. The sectors most severely affected by this attack include online payment services, e-commerce, and social media [2, 3]. Phishing attacks make use of the aesthetic resemblance between fake and legitimate websites [4]. Phishing has taken on several forms throughout history, including legal, educational, and awareness efforts [5]. Phishing attacks use several methods to access sensitive information, including link manipulation, filter evasion, website forgery, covert redirection, and social engineering [6, 7]. It is estimated that internet-based theft, fraud, and exploitation would account for an astonishing $4.2 billion in financial losses in 2020, according to the FBI's Internet Crime Complaint Center 2020 report [8]. During this attack, the attacker mainly produces a website that is identical to the genuine web page in appearance. The phishing web page's URL is subsequently sent to thousands of Internet users through email and other contact forms [9]. Typically, the false email content creates panic, urgency, or promises money in exchange for the recipient taking immediate action. The fake email will prompt customers to change their PIN to prevent their debit/credit card suspension. When a user changes their sensitive credentials inadvertently, cyber thieves get the user's information [10]. Phishing attacks are not just used to get information, they have also become the primary technique for

S. M. Mahamudul Hasan (✉) · N. M. Jakilim · Md. Forhad Rabbi
Shahjalal University of Science and Technology, Sylhet, Bangladesh
e-mail: smmahamudul32@student.sust.edu

N. M. Jakilim
e-mail: nirjas01@student.sust.edu

Md. Forhad Rabbi
e-mail: frabbi-cse@sust.edu

R. M. S. Rahman Pir
Leading University, Sylhet, Bangladesh
e-mail: rumelpir@lus.ac.bd

B. Unhelker et al. (eds.), *Applications of Artificial Intelligence and Machine Learning*,
Lecture Notes in Electrical Engineering 925,
https://doi.org/10.1007/978-981-19-4831-2_48

Fig. 1 Unique phishing activity trends

distributing other kinds of harmful software, such as ransomware. 91% of current cyber-attacks begin with phishing emails [11].

Phishing assaults account for more than half of all cyber fraud affecting Internet users. More than 245,771 distinct phishing websites were identified in January 2021, according to the APWG study. Monthly attack growth rose by 1477% during a ten-year period from 2010 to 2020. (363661 Phishing attacks in 2010 and an average of 5371508 attacks in 2020). From 2010 through 2020, Fig. 1 depicts the rise of phishing assaults [12].

Numerous variations of phishing attacks have occurred throughout history [13]. The attacker may be motivated by identity theft, financial gain, or celebrity. Scientists and researchers face significant challenges when it comes to identifying and blocking phishing attacks [14]. Even the most seasoned and informed users may face an assault. Phishing attacks usually include the transmission of a fake email purporting to be from a reputable company or organization, requesting sensitive information such as a bank login or password. Phishing communications are delivered through email, SMS, instant messaging, social media, and voice over internet protocol (VoIP) [15]. However, the most frequent form of attack is through email. 65% of phishing efforts include the use of malicious URLs in emails [16]. Due to the fact that phishing websites are only up for a limited time, the phisher may flee immediately after committing the crime. That's why automated systems are needed to prevent them as soon as possible. To determine if a website is phishing or not, a variety of methods have been developed. Attackers may use several methods at different phases of the attack cycle. Among other things, network security, user education, user authentication, server-side filters, client-side tools, and classifiers are some of the methods that are available. While each kind of Phishing attack is unique, the bulk of them

have certain characteristics and patterns [17]. Due to the essential role of machine learning techniques for identifying patterns in data, it has become possible to identify a large number of common Phishing characteristics, as well as to recognize Phishing websites [18]. Specifically, the purpose of this paper is to examine and evaluate a number of machine learning methods for identifying fake websites. Logistic Regression, Random Forest, Support Vector Machine, KNN, Naive Bayes, and the XGBoost classifier were among the machine learning methods we investigated.

The remainder of this research paper is divided into the following sections: The second section covers similar studies on phishing websites. Section 3 explored our suggested method of work. In Sect. 4, we provide a short description of the dataset. In Sect. 5, we examined different machine learning techniques to detect phishing and summarized the experiments and their outcomes. The conclusion and future study are described in Sect. 6.

2 Related Work

There are various methods of phishing detection and they are list-based and machine-learning-based etc. The most often used detection technique is list-based. Whitelists consist of legitimate websites, while blacklists consist of phishing websites. Phishing detection systems that are list-based rely on these lists to identify phishing attempts. C Whittaker, B Ryner, M Nazif [19] compiled and published a whitelist of all URLs accessed through the Login user interface. When a user visits a website, the system informs the user if their data is incompatible with the site. This method may be used by a user for the first time when they visit an authorized website. SL Pfleeger, G Bloom [20] developed an automatic whitelist of user-approved websites. They used two stages of feature extraction, which are domain-IP address matching and source code connections. They got a true positive rate of 86.02% for all observations, whereas false negatives accounted for 1.48%. Zhang et al. [21] identified phishing by developing CANTINA that uses TF-IDF techniques. It is a content-based approach. The keywords are put into Google. When a website appears in search results, it earns the confidence of users. CANTINA Plus is equipped with fifteen HTML-based features. Despite the algorithm's 92% accuracy rating, it can give a substantial number of false positives. Islam et al. [22] created a categorization system for communication by reviewing the website titles and messages. The technique is designed to minimize false positives. The research gathered information on URL-specific characteristics such as length, subdomain names, slashes, and dots. Rule mining was utilized to develop detection rules as a priority. In testing, 93% of phishing URLs were correctly identified.

To categorize phishing attempts more correctly and effectively, an adaptive self-structuring neural network was employed by Rami et al. [23]. It includes seventeen features, some of which are dependent on third-party services. As a result, real-time execution is sluggish. While it has the potential to enhance accuracy, it is not currently used. It manages noisy data by using a small dataset of 1400 elements. Others combine artificial intelligence and image processing. For image/visual-based applications, the internet domain (web history) is needed to recognize phishing attempts. Basit et al. [24] proposed an approach that circumvents these constraints. They categorized features according to whether they included hyperlinks, third-party material, or masked URLs. By using third-party services, the system's accuracy is increased to 99.55%, while detection time and latency are decreased. NLP receives little attention in the scholarly literature (NLP). In a recent study, Peng et al. [25] used natural language processing to detect phishing emails. This program scans the plain text content of emails for harmful intent. It gathers queries and responses via the use of natural language processing (NLP). Phishing attempts are detected using a custom-built blacklist of word pairs. The algorithm was trained on 5009 phishing emails and 5000 real emails prior to becoming public. Their experimental research demonstrated a 95% accuracy rate.

3 Proposed Approach

The method we use to identify phishing websites is one that is based on machine learning. Our model incorporates a variety of machine learning techniques, including logistic regression, KNN, decision trees, Random Forest, support vector machines, and gradient boosting. We collected the dataset from kaggle [26] and highlighted the dataset's vector in our model (Fig. 2).

Then, we utilized this dataset for training six machine learning classification models to identify the characteristics of a phishing website. To implement the machine models, we have used the sci-kit learn library [27]. We also utilized their tweaking parameters to hyparametertune the models to get the best results for a particular model. Finally, we compare the results and find out the best-performing models and evaluate the reasons for their good performances. Our categorization algorithm detects Phishing websites with an accuracy of about 97%. And our model is capable of detecting approximately 97.48% of the genuine phishing sites.

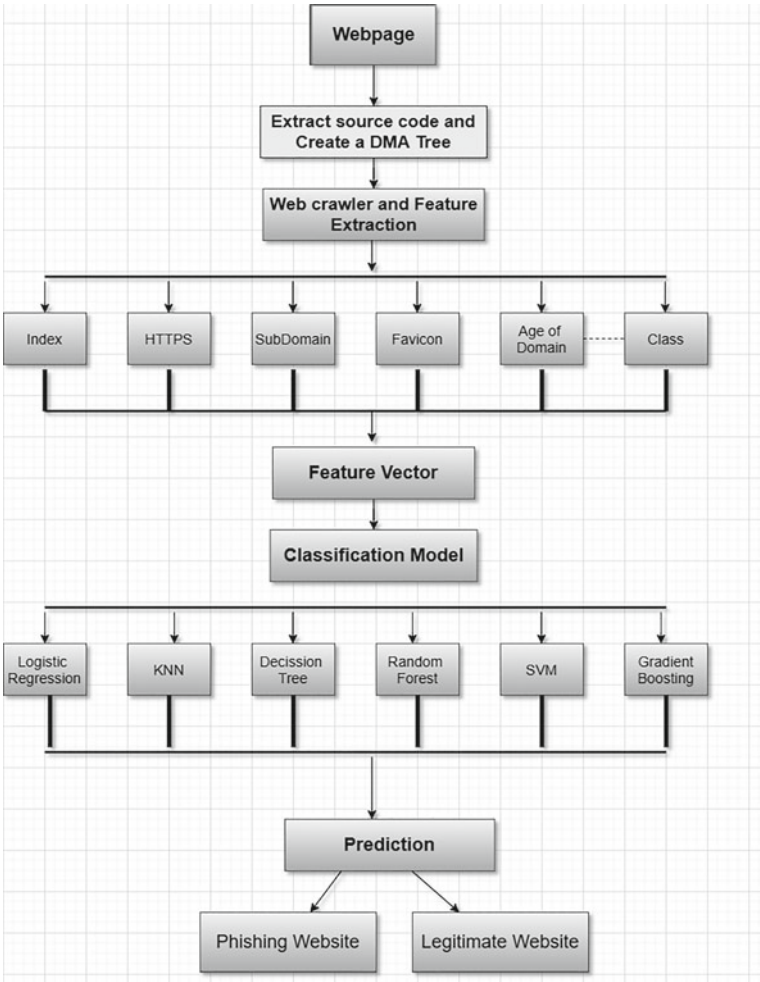


Fig. 2 Diagram of our approach

4 Dataset

One of the most important challenges we faced during our research was the scarcity of phishing databases. There have been a large number of academic papers on the subject of phishing detection. However, none of them have made the datasets used in their research available to the general public. The absence of a common feature set that captures the features of a phishing website also makes it more difficult to collect useful data, which makes it more difficult to build a useable dataset in the first place. Numerous academics carefully examined and benchmarked the dataset used in our analysis. We collected the dataset from kaggle. [26] which contains about

Fig. 3 URL components of a legitimate website

11,054 sample websites with 32 features. Nearly 6080 legitimate websites and 4974 phishing websites are included in the dataset. In our dataset, we give a score of 1 to genuine websites and –1 to phishing websites. We utilized 30% of the samples for testing and 70% for training. Each website is evaluated to see if it is legitimate or fake.

We categorize the 32 features of our dataset into three categories. The following categories are described:

4.1 Address Bar Based Features

The following elements are considered in the address bar-based features. Long URL, Short URL, Symbol@, Redirecting/, Prefix Suffix-, Subdomains, HTTPS, Domain-RegLen, Favicon, NonStdPort, HTTPSDomainURL, Request URL, and Anchor URL are all used in the index. The address bar-based features are often referred to as the link URL-based features. The address bar is described in the following manner (Fig. 3):

4.2 Abnormal Based Features

There are 9 features that define abnormal-based characteristics. LinksInScriptTags, Server Form Handler, Info Email, Abnormal URL, Website Forwarding, StatusBar-Cust, Disable Right Click, Using popup Window, and IframeRedirection are some of them.

4.3 Domain Based Features

There are 8 features that describe domain-based features. These are the following: Domain Age, DNSRecording, Website Traffic, Links Pointing to Page, PageRank, Google Index, Status Report, and class.

If the URL-based feature has a value of −1, it is a phishing website. If the value is 0, the website is suspect. If the URL value includes one, it indicates that the website is genuine.

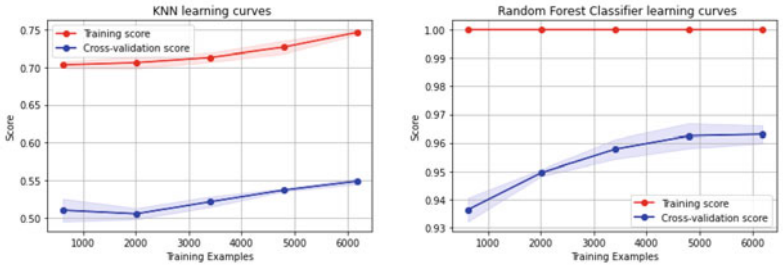
5 Result and Analysis

We utilized 10-fold cross-validation to evaluate the model's overall performance in our trials. 10 sub-samples were drawn from the original data set using a random number generator. Three samples are tested (30%), while the other samples are used to train model-based categorization algorithms. Because phishing detection is categorical in nature, we must employ a binary classification model to identify phishing assaults. "−1" denotes a phishing sample, while "1" denotes a genuine sample. We identified phishing websites using a variety of machine learning models, including logistic regression, random forest, KNN, SVM, gradient boosting, and decision trees.

We assessed these models' accuracy, precision, recall, F1 score, and confusion matrix, and then utilized a variety of feature selection and hyperparameter tweaking techniques to get the best possible results. The precision, recall, and F1 scores and accuracy of different models, as well as their overall performance, are summarized in Table 1. The accuracy of different models, as well as their overall performance, is compared in Fig. 5. The Random Forest has been proven to be extremely accurate, reasonably resistant to noise and outliers, simple to build and comprehend, and capable of implicit feature selection in our studies. In Fig. 4, we show the learning

Table 1 Evaluation of all the models

Algorithms	Precission	Recall	F1 score	Accuracy %
Logistic Regression	0.92	0.93	0.93	92.76
KNN	0.52	0.55	0.54	60.45
Decision Tree	0.95	0.95	0.95	94.75
Random Forest	0.97	0.96	0.97	97.17
SVM	0.50	0.28	0.36	56.04
Gradient Boosting	0.94	0.95	0.95	94.75



(a) Learning curve of Random Forests

(b) Learning curve of KNN

Fig. 4 Learning curves of RF and KNN

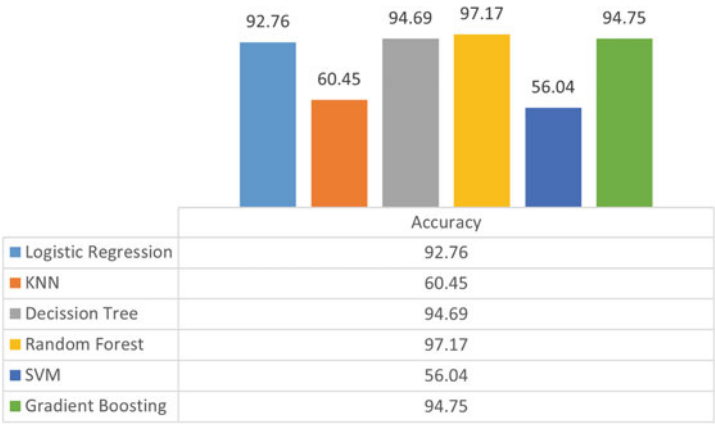


Fig. 5 Accuracies of the models

curves of random forest. The Random Forest offers a number of benefits over the decision tree, the most important being its resistance to noise.

By increasing the number of trees in each woodland inside the forest, the Random Woodland decreases variance. The primary drawback of Random Forests was the large number of hyperparameters that required tuning to attain optimum performance. Additionally, it adds a random aspect to both the training and testing data, which may not be appropriate for all data sets and circumstances. The study could establish the optimal classification accuracy of the KNN by using k = 10. There is no one-size-fits-all k value for KNN classification. In Fig. 4, we show the learning curves of the KNN classifier. Due to the huge number of neighbors, it is computationally costly to develop a solution. Additionally, we discovered that a few neighbors provide the most flexible fit, with low bias but a high variation, while a large number of neighbors produces a smoother decision boundary with a lower variance but greater bias.

Logistic regression is predicted to be 92.76% accurate. Additionally, our system accurately detects about 93.50% of true positives in the confusion matrix. We can evaluate the model's correctness throughout both the training and testing stages by examining the training and cross-validation scores. The actual accuracy of the KNN model is just 55.28%, and the model cannot identify 44.71% of phishing websites. This accuracy is poor, and the total performance of the KNN model is 60.45%. When the accuracy of decision tree tests is considered, the cross-validation score performs well. The decision tree classifier has a true positive score of 95.32%, but a false positive score of just 4.67%. With an overall accuracy of 94.69%, the model is very accurate. The Random forest has the greatest accuracy among all the models. At level 1, the training score of the model is negligible. Cross-validation surpasses single-validation. This model has a true positive rate of 97.48% and a false negative rate of 2.51%. The system works optimally 97.17% of the time. The support vector machine model performs the least well of all the models. The accuracy is optimal 56.0% of the time. It is unable to identify any phishing websites effectively using this. The training score

is the lowest, but the cross-validation score is close to the training score overall. As a result, the model is unable to function properly. In our model, the gradient boosting method works well. It detects true positives at a rate of 95.53% and false positives at a rate of just 4.46%. The model's total accuracy is 94.75%.

The primary benefit of Gradient Boost over other techniques such as decision trees and support vector machines is its speed. Additionally, it has a regularization parameter that significantly lowers variance. To further enhance the generalizability of this approach, the learning rate, and subsamples from features such as random forests are combined with the regularization parameter. Compared to Logistic Regression and Random Forests, Gradient Boost is more difficult to comprehend, visualize, and change. Numerous hyperparameters may be adjusted to improve overall performance. Gradient Boost is an enticing technique to use when both speed and accuracy are required. Despite this, more resources are needed to train the model, since model tweaking takes additional effort and skill on the part of the user to get statistically significant results.

The decision tree outperforms the KNN, Logistic regression, and SVM in our model. Due to the huge volume of data and the diversity of characteristics included therein, the decision tree works well in this scenario. The Decision Tree has two nodes. The Decision Node is the first of these nodes, followed by the Leaf Node. In contrast to decision nodes, which are used to make choices and include many branches, leaf nodes reflect the result of those choices and contain no further branches that branch out to other locations. The judgments or tests are done in light of the dataset's characteristics.

For SVM, we just apply the linear kernel model. Our prior experience indicates that the linear kernel does not perform well on this dataset. Consequently, SVM is ineffective at detecting phishing websites. It is unable to properly detect any phishing websites. Despite the size of our dataset, the SVM technique is not designed for big data sets. When the data set has a high level of noise and the target classes overlap, SVM performs poorly. It is unusual for the SVM to perform poorly when a single data point has more features than there are training data samples. Therefore, SVMs fail in our model.

6 Conclusion

We developed and tested six phishing website classifiers on a dataset of 6080 legitimate websites and 4974 phishing websites in this study. Classifiers such as Logistic Regression, Decision Trees, Support Vector Machines, Random Forests, KNNs, and Gradient Boosting are examined. Our classifiers, Random Forest and Gradient Boost, perform well in terms of computation time and accuracy, as shown in Tables 1. Experimental findings indicate that logistic regression works best for the identification of phishing websites. The suggested method has reasonably high accuracy in identifying phishing websites as it was obtained for random forest classification. It has more than 97.41% true positive rate and 2.58% false positive rate. Moreover, our method's

accuracy, precision, and f1 score are 97.17, 97.80, and 97.59%, respectively. We have also examined the area under the classification model and the learning curve for all the models to discover a better measure of accuracy. Our experiment calculated training and cross-validation scores independently for all classification models used to categorize correct web pages.

Our model could not make use of support vector machines because SVM uses the linear kernel model. That's why when the input is noisy and the target classes overlap, SVM performs poorly. When a single data point has more features than the amount of training data samples, it fails. As a consequence, SVMs don't perform optimally in our model. Utilizing a different SVM kernel may be beneficial. Also, using a polynomial, Sigmoid, or RBF kernel may increase accuracy. Logistic regression is predicted to be 92.76% accurate. The KNN accuracy is poor, and the total performance of the KNN model is 60.45%. When the accuracy of decision tree tests is considered, the cross-validation score performs well. The decision tree classifier has a true positive score of 95.32%, but a false positive score of just 4.67%. With an overall accuracy of 94.69%, the model is very accurate.

References

1. Shaikh AN, Shabut AM, Alamgir Hossain M (2016) A literature review on phishing crime, prevention review and investigation of gaps. In: 2016 10th international conference on software, knowledge, information management & applications (SKIMA). IEEE
2. Scheau C, Arsene A, Dinca G (2016) Phishing and e-commerce: an information security management problem. J Def Resources Manage 7(1):12
3. Sarjiyus O, Oye ND, Baha BY (2019) Improved online security framework for e-banking services in Nigeria: a real world perspective. J Sci Res Rep 1–14
4. Mohammad RM, Thabtah F, McCluskey L (2015) Tutorial and critical analysis of phishing websites methods. Comput Sci Rev 17:1–24
5. Adebowale MA et al (2019) Intelligent web-phishing detection and protection scheme using integrated features of Images, frames and text. Expert Syst Appl 115:300–313
6. Ali A (2016) Social engineering: phishing latest and future techniques. Accessed 10 Mar 2015
7. Goel D, Jain AK (2018) Mobile phishing attacks and defence mechanisms: state of art and open research challenges. Comput Secur 73:519–544
8. FBI releases the internet crime complaint center 2020 internet crime report, including COVID-19 scam statistics. https://www.fbi.gov/news/pressrel/press-releases/fbi-releases-the-interne-crime-complaint-center-2020-internet-crime-report-including-covid-19-scam-statistics
9. Jain AK, Gupta BB (2016) A novel approach to protect against phishing attacks at client side using auto-updated white-list. EURASIP J Inf Secur 2016(1):1–11
10. Dhamija R, Doug Tygar J, Hearst M (2006) Why phishing works. In: Proceedings of the SIGCHI conference on Human Factors in computing systems
11. 91% of all cyber attacks begin with a phishing email to an unexpected victim. https://www2.deloitte.com/my/en/pages/risk/articles/91-percent-of-all-cyber-attacks-begin-with-a-phishing-email-to-an-unexpected-victim.html
12. Phishing activity trends reports. https://apwg.org/trendsreports/
13. Charoen D (2011) Phishing: a field experiment. Int J Comput Sci Secur (IJCSS) 5(2):277
14. Jakobsson M, Myers S (eds) Phishing and countermeasures. Understanding the increasing problem of electronic identity theft. Wiley, Hoboken
15. Ramzan Z (2010) Phishing attacks and countermeasures. In: Handbook of information and communication security, pp 433–448

16. Must-know phishing statistics. https://www.tessian.com/blog/phishing-statistics-2020/
17. Jain AK, Gupta BB (2021) A survey of phishing attack techniques, defence mechanisms and open research challenges. Enterp Inf Syst 1–39
18. Passos IC, Mwangi B, Kapczinski F (2016) Big data analytics and machine learning: 2015 and beyond. Lancet Psychiat 3(1):13–15
19. Whittaker C, Ryner B, Nazif M (2010) Large-scale automatic classification of phishing pages
20. Pfleeger SL, Bloom G (2005) Canning spam: proposed solutions to unwanted email. IEEE Secur Priv 3(2):40–47
21. Zhang Y, Hong JI, Cranor LF (2007) Cantina: a content-based approach to detecting phishing web sites. In: Proceedings of the 16th international conference on World Wide Web
22. Islam R, Abawajy J (2013) A multi-tier phishing detection and filtering approach. J Netw Comput Appl 36(1):324–335
23. Mohammad RM, Thabtah F, McCluskey L (2014) Predicting phishing websites based on self-structuring neural network. Neural Comput Appl 25(2):443–458
24. Basit A et al (2020) A comprehensive survey of AI-enabled phishing attacks detection techniques. Telecommun Syst 1–16
25. Peng T, Harris I, Sawa Y (2018) Detecting phishing attacks using natural language processing and machine learning. In: 2018 IEEE 12th international conference on semantic computing (ICSC). IEEE
26. Phishing website detector. https://www.kaggle.com/eswarchandt/phishing-website-detector
27. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Duchesnay E (2011) Scikit-learn: machine learning in Python. J Mach Learn Res 12:2825–2830