Phishing URL detection using machine learning methods

Hasane Ahammad Shaik
K L University
185 PUBLICATIONS **917** CITATIONS

Advances in Engineering Software 173 (2022) 103288

Contents lists available at ScienceDirect

Advances in Engineering Software

journal homepage: www.elsevier.com/locate/advengsoft

Research paper

Phishing URL detection using machine learning methods

SK Hasane Ahammad[a], Sunil D. Kale[b], Gopal D. Upadhye[b], Sandeep Dwarkanath Pande[c,*], E Venkatesh Babu[a], Amol V. Dhumane[b], Mr. Dilip Kumar Jang Bahadur[d]

[a] *Department of ECE, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur 522502, India*
[b] *Pimpri Chinchwad College of Engineering, Pune 411044*
[c] *MIT, Academy of Engineering, Alandi, Pune, India*
[d] *Department of Computer and Information Sciences, Himalayan School of Science & Technology, Swami Rama Himalayan University, Dehradun, Uttarakhand, India*

ARTICLE INFO

Keywords:
Phishing
Cybercrime
Machine learning
Malicious website
URL
Light GBM

ABSTRACT

Phishing is among the most concerning issues in a constantly changing world. The increasing use of the Internet has led to a new way of stealing data, known as cybercrime. Cybercrime refers to stealing private information and violating privacy through computers. The primary technique used is phishing. Phishing via URLs (Uniform Resource Locators) is one of the most common types, and its primary goal is to steal the data from the user when the user accesses the malicious website. Detecting a malicious URL is a significant challenge. This work aims to provide a solution for detecting such websites with the help of machine learning algorithms focused on the behaviors and qualities of the suggested URL. The web security community has created blacklisting services to identify malicious websites. A variety of methods, such as manual reporting, and site analysis heuristics are used to create these blacklists. Due to their recentness, lack of evaluation, or incorrect evaluation, many malicious websites inadvertently escape blacklisting. To create a machine learning model for detecting whether a URL is malicious or not, algorithms such as Random Forests, Decision Trees, Light GBM, Logistic Regression, and Support Vector Machine (SVM) are used. Extracting features is the first step, and applying the model is the next step.

1. Introduction

Due to the rise of internet usage over the past few years, more and more people are using the Internet as a platform to make online transactions and share their information and e-commerce. As the Internet increased, a new form of crime was created, known as cybercrime. There are many ways for cybercriminals to steal information, and most of them use phishing to accomplish this. There are various types of phishing, including vishing, spear phishing, whaling, and email phishing. In 1990, phishing was first reported, and it was used to steal passwords. There has been an increase in phishing attacks in recent years. URL phishing is one such attack.

A URL is a website address that represents the location of a website on a network and the means of gaining access to it. By accessing the URL, we connect to the database on the server, which stores all the details relating to the website, and it contains a webpage that displays them. URLs are divided into two categories: malicious and benign. Malicious URLs are used in URL phishing, while benign URLs are harmless and secured. A cybercriminal will create a site that looks like the real one, and all of its information will be identical to that of the absolute URL. The URL will appear as an advertisement on other websites, and the fraud will happen when the user enters their credentials. And another way is by sending the malicious URL to the user through email, and when the user tries to open the URL some nasty virus will be downloaded, this allows the cybercriminals to access the information to commit their crimes. Malicious and benign URLs look similar, so to distinguish them we need to extract some features from them. Detecting malicious URLs requires extracting some of their features from them, then comparing these features to determine whether the URL is malicious or benign.

2. Literature review

There have been studies conducted out many theories and methods have been proposed by various authors to detect phishing URLs, and one of the theories is to detect whether the URL is malicious (or not) by using

* Corresponding author.
E-mail address: sandeep7887pande@gmail.com (S.D. Pande).

https://doi.org/10.1016/j.advengsoft.2022.103288
Received 27 July 2022; Received in revised form 19 August 2022; Accepted 9 September 2022

features based on the weightage of the message content.

Carolin and Rajsingh [1] proposed a model to detect malicious URLs is to use associate rule mining, a process in which data mining is performed. Data mining, it is the extraction and organization of information from a dataset [1]. Taking both malicious and legitimate URLs, he conducted a study to determine how the features of the URL change for malicious and legitimate URLs. By conducting this study, he provided a brief overview of the URL features by conducting a study that took both malicious and legitimate URLs.

Garera [2] proposed technique is the use of features such as page ranking, whether the URL's domain is present in a white domain table, and word analysis, which looks for which words are used more in malicious URLs [2]. Using this data, a machine learning model was developed that could detect malicious URLs [2].

Mohammed et al. [3] proposed a model in which a machine learning model was developed by using results generated by Microsoft Reputation Services, as well as other URL based features. By using this model, we can determine whether a URL has malicious intent. The model provided accurate results. Microsoft Reputation Services is a tool developed by Microsoft which provides URL classification to protect against malware.

Blum et al. [4] presented a model in which he asserted that we could detect whether a URL is malicious or not without looking at web page contents by using URL based characteristics, For example, the domain name, An occurrence of special characters, etc. A machine learning model was developed using all of these features.

There were some models proposed to detect whether URL is malicious or legitimate The CANTINA method, which is based on HTML features, and the advanced version of CANTINA which adds a few new features. An alternative method is using NLP algorithms, where a word dictionary for malicious as well as benign URLs is created, including all word-based features, and This information is used to create a machine learning model for detecting malicious URLs.

Parekh et al. [5] proposed a method to detect the malicious website using document object model features. Programming languages like XML and HTML use the document object model as an API, the document object model represents the HTML or XML code in a tree structure, and the tree includes features such as gray histograms, color histograms, and spatial relationships that can be used to detect phishing URLs.

Additionally, Pradeepthi and Kannan [6] presented a visual method for identifying malicious websites. In this work detection of phishing involves looking at text pieces and styles, as well as images found on the webpage.

A study by Fu [7] shed light on PhoneyC, a virtual honey pot that is used to study the nature of malicious URLs that cybercriminals use to steal information.

We use the EMD to calculate the signature distances of the Web page images based on the technique proposed by Sahoo [8]. They converted the webpages to images and then used the feature such as color to determine the image sign. In some studies, it has also been demonstrated that malicious URLs can be detected by checking the relationship with previously used domains.

Another feature of malicious URL detection is based on the HTML features, so in this study, they proposed a way to check if there is any malicious content in the URL using the beautifulsoup Python package used to parse HTML and XML files and based on that, we can detect the malicious URL. As another alternative, string-based algorithms can be used where the URLs can be preprocessed in such a way that both malicious and legitimate URLs will have a word cloud, but here the word cloud merely consists of what words are most common in legitimate URLs, and what words are most common in malicious URLs, and then based on the word clouds based on the analysis of the malicious and legitimate word clouds. With machine learning algorithms, we can determine whether a URL is malicious or legitimate.

3. Methodology

3.1. Dataset

In this dataset, there are 3000 URLs, including 1500 malicious URL data and 1500 benign URL data. Phishing URLs were collected from a service called Phish Tank, an open-source service. Phish Tank provides collaborative data on phishing on the Internet through a database of phishing information, with the service providing multiple formats of data such as csv, json, and many more, which are updated hourly. My research led me to find a data set that contains benign, spam, phishing, malware & defacement URLs. My source is the University of New Brunswick and the number of legitimate URLs in this collection is 35,300. Malicious URLs and benign URLs are combined in this data set.

The next step after collecting the dataset is:

a Data preprocessing, including merging the data and a major challenge when attempting to add a dataset to the machine learning model is null values. Because of this, all null values are removed before adding the dataset to the machine learning model.

b Feature extraction, in this process, lexical domain-based features from the final dataset are extracted using Python modules such as urlparse and whois.

c Finally, we apply the machine learning model to all the features generated by the feature extraction module with the help of machine learning algorithms such as Random Forest Classifier, Decision Tree, and Light GBM algorithms.

3.2. Feature extraction

As part of this step, we extract features from the URL dataset. The extracted features are classified into Address Bar based Features and Domain based Features and a total of 15 features are taken into consideration.

- Domain name:

  Currently, we are just extracting the domain present in the URL. This feature is not particularly useful during training. It may even be dropped entirely during training.

- Have IP:

  In usual URLs, we do not see any IP address, but there will be a domain name. If there is an IP address in the URL, then we could conclude that the URL is malicious. Cybercriminals use IP addresses in URLs to steal sensitive information. It is a malicious URL if the IP address exists in URL, and it will be assigned a 1 otherwise a 0 as it is a benign URL.

- Have @ symbol:

  URLs with '@' symbols have a value of 1 (phishing) or 0 (legitimate).

- Length and Depth of URL:

  A long URL is often used by cybercriminals to hide the anonymous part, so URLs longer than 54 characters are assigned 1 (phishing) or 0 (benign) and depth of the URL is nothing but how many subpages it contains.

- Position of '//' in the URL:

  The "//" should be present at the 6th position if the URL starts with HTTP, or at the 7th position if the URL starts with HTTPS. If "//" is found anywhere else, then the value for this feature should be 1 (phishing) or 0 (benign).

- HTTP/HTTPS in Domain name:

The value assigned to this feature is 1 (phishing) or 0 (benign) depending on whether the URL includes "http/https" in the domain part.

- Prefix/ Suffix '- 'in the Domain:

Although URLs do not contain "-" cybercriminals may add the "-" to the URL, so a value of 1 would indicate phishing or a value of 0 would indicate benign URL.

- Statistical Report:

Several companies, including Phish Tank and Stop Bad ware, are generating multiple statistical reports on phishing websites every month or every quarter. If host belongs to top phishing domains, then the value assigned to this feature is 1 (phishing) or 0 (benign).

- DNS Record:

WHOIS is a registry that contains information about domain names such as registration and contact details. If there is no DNS record, then the value assigned to this feature is 1 (phishing) or 0 (benign).

- Domain Based Features:

Domain-based features include features such as web traffic, domain age, domain end period, and subdomains. The web traffic is the number of visitors visited a URL or webpage, and it is derived from the Alexa database. If a URLs rank is under 100,000, then this feature has a value of 1 (phishing) or 0 (benign). The age of the domain is important because the age of a malicious URL is less than 12, so if the age of domain is less than 12, this feature will have a value of 1 (phishing) or 0 (benign). Domain end period, in this feature, we assign a value of either 1 (phishing) or 0 (benign) depending on whether the difference between expired time and the current time of a domain is less than 6 months.

- Sub-domain:

Whenever the "." count in the URL is greater than 3, that website is malicious, and the value assigned to it is 1 (phishing) or 0 (benign).

3.3. Machine learning algorithms

- Decision tree algorithm:

An improved version of classification and regression trees is the decision tree algorithm.

For tasks such as classification and regression, decision trees are commonly used. The idea behind a decision tree is to determine a decision by asking if and else questions. The idea is to learn what frequency of if and else questions leads us to the correct answer quickly. These questions are called tests in machine learning and called as leaf. The algorithm searches over all possible tests to obtain the most informative tree about the target variable. The problem statement here is a classification related so the decision tree algorithm uses Entropy and Gini impurity.

$$L_H = \sum_{i=1}^{c} f_i(1-f_i) \tag{1}$$

C is the number of unique labels and f is the frequency at that point.

$$L_G = -f_i \log(f_i) \tag{2}$$

- Random forest algorithm:

Random forest algorithm is also used for both classification and regression-related problems. In any classification and regression problem, a random forest algorithm is used. A random forest is nothing more than a collection of decision trees, so for regression problems, the output will be an average of the decision trees. Similarly, for classification-related problems, the output will be the most common result derived from all the decision trees.

$$RFfi_i = \frac{\sum_{j \in alltrees} normfi_{ij}}{\mathrm{T}} \tag{3}$$

For all the decision trees, feature importance will be calculated, and the average sum of all calculated feature importance will be used.

$$fi_i = \sum_{j:nodes\ j\ splits\ on\ feature\ i} s_j C_j \tag{4}$$

fi sub i is Feature i: its significance and s sub j is Samples that reach node j.

- Light GBM:

Light GBM is a framework based on decision trees that uses GOSS or Gradient-based one-side sampling and EFB or Exclusive Feature bundling. It is used for increasing the accuracy of the algorithm as well as improving memory usage. In light GBM, data and features are down-sampled, using GOSS and EFB, to reduce complexity of the histogram building process. As it is based on decision trees, it uses trees and splits the trees based on leaf unlike other boosting algorithms where the tree grows level by level and the leaf does not change so there is less loss than other boosting algorithms.

- Logistic regression

The most used statistical model for predicting binary data in various disciplines is logistic regression. Its ease of use and excellent interpretability have led to its widespread application. It often makes use of the logit function as a component of generalized linear models.

$$log \frac{M(a;\alpha)}{1-M(;\alpha)} = \alpha^T a \tag{5}$$

where a is a vector of M predictors $a = (a1, a2, \ldots.., a_M)$, and a is a $M \times 1$ vector of regression parameters.

When the relationship between the data is roughly linear, logistic regression works well. However, if there are intricate nonlinear interactions between the variables, it performs badly. Additionally, compared to other strategies, it requires more statistical assumptions before use. Additionally, if there are missing data in the data set, the prediction rate is impacted.

- SVM

SVM is a machine learning method based on supervised learning that may be used for both classification and regression. Based on its strong foundation in statistical learning theory and the positive results obtained in several sectors of data mining challenges, the SVM is considered a new approach that is quickly gaining favor. SVM is a statistical learning-based classification approach that has been effectively applied in several nonlinear classification applications involving big datasets and problems. Every hyper-plane is determined by its direction (*a*), the precise position in space or a threshold is (*b*), (*lxi*) denotes the input array of constituent N and indicates the category. Eqs. (6) and (7) show a collection of the training cases.

$$(lx1, y1), (lx2, y2), \ldots.., (lxm, ym); lxi \in R^{ds} \tag{6}$$

m stands for the number of training datasets, and ds stands for the number of input dataset dimensions. The following is a description of the function of decision:

$$f(lx,\ a, b) = sgn((a.lxi) + b), a \in R^{ds},\ b \in R \tag{7}$$

Utilizing the SVM for system training has several benefits, one of which is its capacity to handle multi-dimensional data. SVM is a classifier that outputs an ideal hyperplane that categorizes new examples from input labelled training data. By maximizing the margin, SVM creates a hyperplane between data sets, as seen in Fig. 1.

4. Results

Therefore, a machine learning model can be created with the help of all the algorithms discussed above, and for testing and training, the machine learning model and 80% of the dataset were used for training and 20% for testing. Light GBM, Random Forest, Decision Tree, Logistic Regression, and SVM are the machine learning methods used to analyze to determine whether such a URL is fraudulent or not. As a result of fitting the dataset to all algorithms, Light GBM produced good results and all performance analysis is provided in Table 1. Light GBM Model has 0.895 in training accuracy and the test accuracy achieves 0.860 also Random Forest achieves 0.883 in training accuracy and the test accuracy holds 0.853. In addition, the Decision tree achieves 0.880 in training accuracy and the test accuracy holds at 0.850.

Figs. 2 and 3 shows a graph of the importance of the different features taken into account as compared to each other. Though there are 15 features, only some are important in increasing accuracy. Random Forest obtains 0.883 in training accuracy and remains 0.853 in test accuracy, Additionally, the decision tree's test accuracy remains at 0.850 while its training accuracy is at 0.880. Logistic Regression obtains 0.878 in training accuracy and remains 0.842 in test accuracy. Also, SVM achieves 0.871 while its training accuracy is at 0.835, whereas Light GBM Model achieves higher i.e., 0.895 in training accuracy and 0.860 in test accuracy.

As you can see from the graph, light GBM has better training and testing accuracy than random forest and decision tree algorithms. The graph represents the descending order of the testing and training accuracy of the algorithm used to predict malicious URLs.

Figs. 4–6 depict the validation curves for all algorithms that were used. The validation curve represents the score, or accuracy, of the model for different values of the hyperparameter of that algorithm.

We can see from the Fig. 4 that both the training and cross validation scores are similar and are increasing gradually, so we can say that the model is performing well.

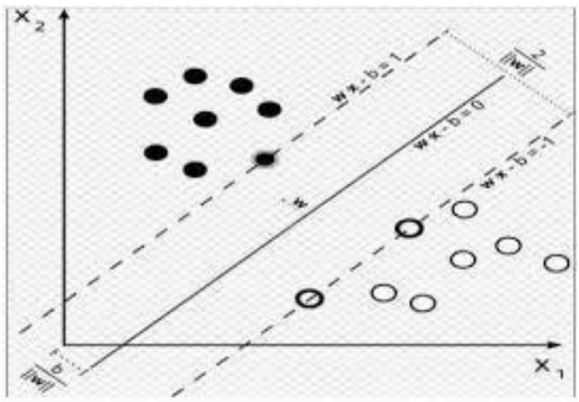

Fig. 1. SVM for classifying phishing websites.

Table 1
Results.

S.NO	ML MODEL	TRAIN ACCURACY	TEST ACCURACY
1	LightGBM	0.895	0.860
2	Random Forest	0.883	0.853
3	Decision Tree	0.880	0.850
4	Logistic Regression	0.878	0.842
5	SVM	0.871	0.835

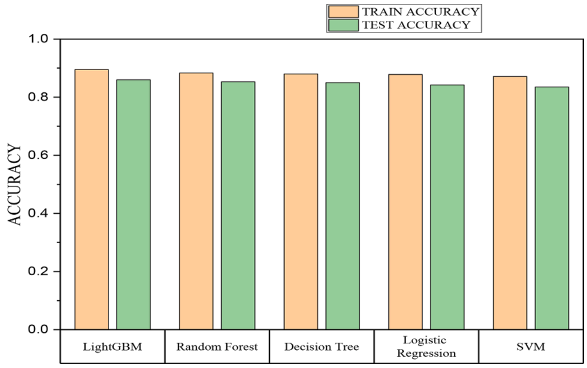

Fig. 2. Accuracy scores for algorithms.

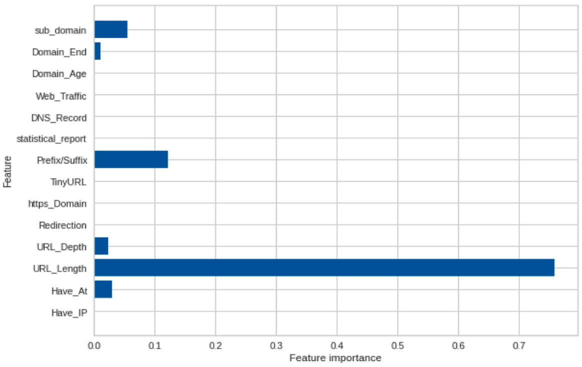

Fig. 3. Feature importance.

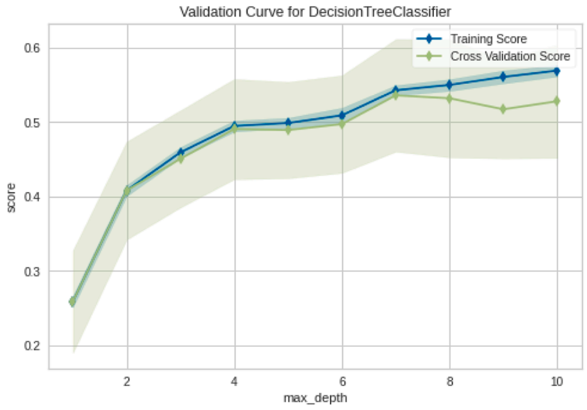

Fig. 4. Validation curve for decision tree classifier.

Fig. 5 also shows that both the training and cross-validation score are similar and are increasing over time, so this model is also performing well.

LGBM is a tree-based algorithm that computes data more quickly

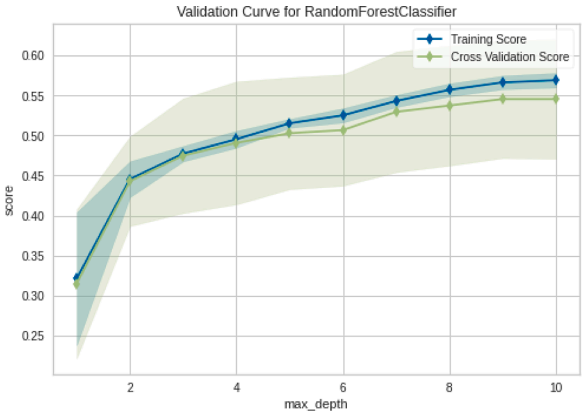

Fig. 5. Validation curve for random forest classifier.

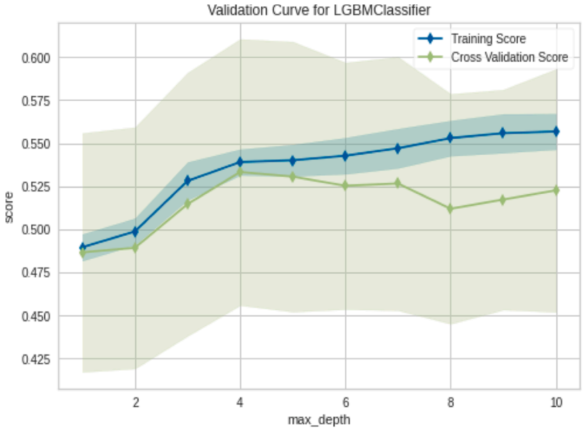

Fig. 6. Validation curve for LGBM classifier.

than other algorithms. Its root and leaf can grow either vertically or horizontally because the algorithm is based on a tree. Fig. 6 shows the validation curve for the Light GBM framework, and both the training and cross validation scores are increasing steadily up to a max depth of six, after which there is a slight deviation for cross validation. We can conclude that the model performs well at max depth 5.

5. Conclusion and future work

Internet users are at serious risk from phishing. Web security faces a significant issue as a result of the rapid development and spread of phishing methods. It is difficult to identify a fraudulent URL, but machine learning methods can help. In this study, using the Random Forest, Decision Tree, Light GBM, Logistic Regression, and Support Vector Machine, we investigated the linguistic and domain-based properties of the URL and created a machine learning model. The Light GBM algorithm produced the best outcomes out of the bunch. However, some of the URLs in the dataset are not in the whois database, thus we are unable to collect all of the features; as a result, we need to add more features and fresh URL data in order to enhance accuracy. We may use our machine learning model to build a search engine in the future, which will enable us to identify any fraudulent URLs and ban them, eliminating phishing also develop a framework that can discover new phishing attack types on its own by giving the surveillance process a more advanced feature.

Author statement

Not Applicable.

Declaration of Competing Interest

The authors declare that we have no conflict of interest.

References

[1] Carolin Jeeva S, Rajsingh EB. Intelligent phishing URL detection using association rule mining. Hum Centr Comput Inf Sci 2022. https://doi.org/10.1186/s13673-016-0064-3.

[2] Garera, S., Provos, N., Chew, M., Rubin, A.D. (2007, November). A framework for detection and measurement of phishing attacks. In Proceedings of the 2007 ACM workshop on Recurring malcode (pp. 1-8).

[3] Mohammed Nazim Feroz SM. Phishing URL detection using URL ranking. In: Proceedings of the IEEE international congress on big data (BigData congress); 2015. https://doi.org/10.1109/BigDataCongress.2015.97.

[4] Blum A, Wardman B, Solorio T, Warner G. Lexical feature based phishing URL detection using online learning. In: Proceedings of the 3rd ACM workshop on security and artificial intelligence, AISEC; 2010. https://doi.org/10.1145/1866423.1866434. 2010 October 8.

[5] Parekh Shraddha, Parikh Dhwanil, Kotak Srushti, Sankhe Smita. A new method for detection of phishing websites: URL detection. IEEE; 2018. p. 949–52.

[6] K.V. Pradeepthi, A. Kannan "Performance study of classification techniques for phishing URL detection", https://ieeexplore.ieee.org/document/7229761, 2022.

[7] A.Y. Fu, "Detecting phishing web pages with visual similarity assessment based on earth mover's distance (EMD)", 2022, 10.1109/TDSC.2006.50.

[8] D. Sahoo, "Malicious URL detection using machine learning: a survey", 2022.

Further reading

[1] The documentation for the Learning curve function, which visualizer wraps, "https://www.scikit-yb.org/en/latest/api/model_selection/validation_curve.html".