IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 7, July 2021

DOI 10.17148/IJARCCE.2021.10734

Detection of Phishing Websites Using Machine Learning

Akash C M[1], Suriya Raman S S[1], Venkatesh N[1], Mrs.S.Kavitha[2]

B.E, Computer Science Engineering, Velammal College of Engg and Tech, Madurai, India[1]

Assistant Professor, Dept. of CSE., Velammal College of Engineering and Technology, Viraganoor, Madurai, India[2]

Abstract: Phishing is a form of fraud in which the attacker tries to learn sensitive information such as login credentials or account information by sending as a reputable entity or person in email or other communication channels. Typically, a victim receives a message that appears to have been sent by a known contact or organization. The message contains malicious software targeting the user's computer or has links to direct victims to malicious websites in order to trick them into divulging personal and financial information, such as passwords, account IDs or credit card details. The aim of this research is to develop these methods of defense utilizing various approaches to categorize websites. Specifically, we have developed a system that uses machine learning techniques to classify websites based on their URL. We have used two classifiers: The Decision Tree classifier and Naive Bayesian classifier. These classifiers were trained and tested using a dataset that has values of features of 88647 websites. The accuracy of the proposed system is 95.4 % which is higher than most of the other proposed systems.

Keywords: Phishing sites; Machine learning; Classification; Malicious sites; Detection; Cybersecurity;

I. INTRODUCTION

Scammers use email or text messages to trick you into giving them your personal information. They may try to steal your passwords, account numbers, or Social Security numbers. If they get that information, they could gain access to your email, bank, or other accounts. Scammers launch thousands of phishing attacks like these every day — and they're often successful. The FBI's Internet Crime Complaint Center reported that people lost $57 million to phishing schemes in one year. Scammers often update their tactics, but there are some signs that will help you recognize a phishing email or text message. **Phishing emails and text messages may look like they're from a company you know or trust.** They may look like they're from a bank, a credit card company, a social networking site, an online payment website or app, or an online store.

One general approach to recognizing illegitimate phishing websites relies on their Uniform Resource Locators (URLs). A URL is a global address of a document in the World Wide Web, and it serves as the primary means to locate a document on the Internet. Even in cases where the content of websites are duplicated, the URLs could still be used to distinguish real sites from imposters. One solution approach is to use a blacklist of malicious URLs developed by anti-virus groups. The problem with this approach is that the blacklist cannot be exhaustive because new malicious URLs keep cropping up continuously. Thus, approaches are needed that can automatically classify a new, previously unseen URL as either a phishing site or a legitimate one. Such solutions are typically machine-learning based approaches where a system can categorize new phishing sites through a model developed using training sets of known attacks.

One of the main problems with developing machine learning based approaches for this problem is that very few training data sets containing phishing URLs are available in the public domain. As a result, studies are needed that evaluate the effectiveness of machine-learning approaches based on the data sets that do exist. This work aims to contribute to this need. Specifically, the goal of this research is to compare the performance of the commonly used machine learning algorithms on the same phishing data set.

In this work, we use a data set, where features from the data URLs have already been extracted, and the class labels are available. We have tested common machine learning algorithms for the purpose of classifying URLs such as the Decision Tree classifier and Naives Bayesian classifier. The remainder of this paper is structured as follows. Section II describes the related work in classifying phishing URLs. Section III provides the details of the data set and methodology. Section IV describes the results of the tests and provides discussion. Section V describes limitations of the present work and directions for the future work.

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 10, Issue 7, July 2021
DOI 10.17148/IJARCCE.2021.10734

II. RELATED WORK

Machine learning techniques that identify phishing URLs typically evaluate a URL based on some feature or set of features extracted from it. There are two general types of features that can be extracted from URLs, namely host-based features and lexical features. Host based features describe characteristics of the website, such as where it is located, who manages it, and when was the site installed. Alternatively, lexical features describe textual properties of the URL. Since URLs are simply text strings that can be divided into subparts including the protocol, hostname, and path, a system can assess a site's legitimacy based on any combination of those components

Many machine learning techniques have been used for detection of malicious URLs. Sadeh et al. [2] proposed a system called PILFER for classifying phishing URLs. They extracted a set of ten features that are specifically designed to highlight deceptive methods used to fool users. The data set consists of approximately 860 phishing e-mails and 6950 non-phishing emails. They used a Support Vector Machine (SVM) as a classifier in the implementation. They trained and tested the classifier using 10-fold cross validation and obtained 92 percent accuracy.

Ma et al. [3] considered the URL classification problem as a binary classification problem and built a URL classification system that processes a live feed of labeled URLs. It also collects URL features in real time from a large Web mail provider. They used both lexical and host-based features. from the gathered features and labels, they were able to train an online classifier using a Confidence Weighted (CW) algorithm. Parkait et al. [4] provide a comprehensive literature review after analyzing 358 research papers in the area of phishing counter measures and their effectiveness. They classified anti-phishing approaches into eight groups and highlighted advanced anti-phishing methods.

Abdelhamid et al. [5] built a system for detecting phishing URLs called Multi-label Classifier based on Associative Classification (MCAC). They used sixteen features and classified URLs into three classes: phishing, legitimate, and suspicious. The MCAC is a rule-based algorithm where multiple label rules are extracted from the phishing data set. Patil and Patil [6] provided a brief overview of various forms of web-page attacks in their survey on malicious webpages detection techniques.

Hadi et al. [7] used the Fast-Associative Classification Algorithm (FACA) for classifying phishing URLs. FACA works by discovering all frequent rule item sets and building a model for classification. They investigated a data set consisting of 11,055 websites with two classes, legitimate and phishing. The data set contained thirty features. They used the minimum support and the minimum confidence threshold values as two percent and fifty percent, respectively.

Nepali and Wang [8] proposed a novel approach to detect malicious URLs using only visible features from social networks. Kuyama et al [9] proposed a method for identifying the Command-and-Control server (C&C server) by using supervised learning and features points obtained from WHOIS and DNS information. They evaluated domain names and email addresses from the WHOIS as input values for machine learning.

In addition to the above solutions, several researchers have surveyed the field of malicious URL detection. Sahoo et al. [10] provide a comprehensive survey and structural understanding of malicious URL detection techniques using machine learning.

III. METHODOLOGY

A. Dataset

The data set used in this paper was downloaded from the results of the paper 'Datasets for phishing websites detection ' by **GregaVrbančič, IztokFisterJr. and ViliPodgorelec.[11].** The dataset in total features 111 attributes excluding the target phishing attribute, which denotes whether the particular instance is legitimate (value 0) or phishing (value 1).

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 7, July 2021

DOI 10.17148/IJARCCE.2021.10734

Table 1. Dataset attributes based on URL.

Nr.	Attribute	Format
1	qty_dot_url	Number of ”.” signs
2	qty_hyphen_url	Number of ”-” signs
3	qty_underline_url	Number of ”_” signs
4	qty_slash_url	Number of ”/” signs
5	qty_questionmark_url	Number of ”?” signs
6	qty_equal_url	Number of ”=” sings
7	qty_at_url	Number of ”@” signs
8	qty_and_url	Number of ”&” signs
9	qty_exclamation_url	Number of ”!” signs
10	qty_space_url	Number of ” ” signs
11	qty_tilde_url	Number of ”˜”signs
12	qty_comma_url	Number of ”,” signs
13	qty_plus_url	Number of ”+” signs
14	qty_asterisk_url	Number of ”*” signs
15	qty_hashtag_url	Number of ”#” signs
16	qty_dollar_url	Number of ”$” signs
17	qty_percent_url	Number of ”%” signs
18	qty_tld_url	TLD char length
19	length_url	Number of characters
20	email_in_url	Is email present

Table 2. Dataset attributes based on domain URL.

Nr.	Attribute	Format
1	qty_dot_domain	Number of ”.” signs
2	qty_hyphen_domain	Number of ”-” signs
3	qty_underline_domain	Number of ”_” signs
4	qty_slash_domain	Number of ”/” signs
5	qty_questionmark_domain	Number of ”?” signs
6	qty_equal_domain	Number of ”=” signs
7	qty_at_domain	Number of ”@” signs
8	qty_and_domain	Number of ”&” signs
9	qty_exclamation_domain	Number of ”!” signs

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 7, July 2021

DOI 10.17148/IJARCCE.2021.10734

Nr.	Attribute	Format
10	qty_space_domain	Number of " " signs
11	qty_tilde_domain	Number of "signs
12	qty_comma_domain	Number of "," signs
13	qty_plus_domain	Number of "+" signs
14	qty_asterisk_domain	Number of "*" signs
15	qty_hashtag_domain	Number of "#" signs
16	qty_dollar_domain	Number of "$" signs
17	qty_percent_domain	Number of "%" signs
18	qty_vowels_domain	Number of vowels
19	domain_length	Number of domain characters
20	domain_in_ip	URL domain as IP
21	server_client_domain	"server" or "client" in domain

Table 3. Dataset attributes based on URL directory.

Nr.	Attribute	Format
1	qty_dot_directory	Number of "." signs
2	qty_hyphen_directory	Number of "-" signs
3	qty_underline_directory	Number of "_" signs
4	qty_slash_directory	Number of "/" signs
5	qty_questionmark_directory	Number of "?" signs
6	qty_equal_directory	Number of "=" signs
7	qty_at_directory	Number of "@" signs
8	qty_and_directory	Number of "&" signs
9	qty_exclamation_directory	Number of "!" signs
10	qty_space_directory	Number of " " signs
11	qty_tilde_directory	Number of "signs
12	qty_comma_directory	Number of "," signs
13	qty_plus_directory	Number of "+" signs
14	qty_asterisk_directory	Number of "*" signs
15	qty_hashtag_directory	Number of "#" signs
16	qty_dollar_directory	Number of "$" signs
17	qty_percent_directory	Number of "%" signs
18	directory_length	Directory char count

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 7, July 2021

DOI 10.17148/IJARCCE.2021.10734

Table 4. Dataset attributes based on URL file name.

Nr.	Attribute	Format
1	qty_dot_file	Number of ”.” signs
2	qty_hyphen_file	Number of ”-” signs
3	qty_underline_file	Number of ”_” signs
4	qty_slash_file	Number of ”/” signs
5	qty_questionmark_file	Number of ”?” signs
6	qty_equal_file	Number of ”=” signs
7	qty_at_file	Number of ”@” signs
8	qty_and_file	Number of ”&” signs
9	qty_exclamation_file	Number of ”!” signs
10	qty_space_file	Number of ” ” signs
11	qty_tilde_file	Number of ”signs
12	qty_comma_file	Number of ”,” signs
13	qty_plus_file	Number of ”+” signs
14	qty_asterisk_file	Number of ”*” signs
15	qty_hashtag_file	Number of ”#” signs
16	qty_dollar_file	Number of ”$” signs
17	qty_percent_file	Number of ”%” signs
18	file_length	file name char length

Table 5. Dataset attributes based on URL parameters.

Nr.	Attribute	Format
1	qty_dot_params	Number of ”.” signs
2	qty_hyphen_params	Number of ”-” signs
3	qty_underline_params	Number of ”_” signs
4	qty_slash_params	Number of ”/” signs
5	qty_questionmark_params	Number of ”?” signs
6	qty_equal_params	Number of ”=” signs
7	qty_at_params	Number of ”@” signs
8	qty_and_params	Number of ”&” signs
9	qty_exclamation_params	Number of ”!” signs
10	qty_space_params	Number of ” ” signs
11	qty_tilde_params	Number of ”signs

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 7, July 2021

DOI 10.17148/IJARCCE.2021.10734

Nr.	Attribute	Format
12	qty_comma_params	Number of ”,” signs
13	qty_plus_params	Number of ”+” signs
14	qty_asterisk_params	Number of ”*” signs
15	qty_hashtag_params	Number of ”#” signs
16	qty_dollar_params	Number of ”$” signs
17	qty_percent_params	Number of ”%” signs
18	params_length	Number of parameters characters
19	tld_present_params	TLD1present in parameters
20	qty_params	Number of parameters

Table 6. Dataset attributes based on resolving URL and external services.

Nr.	Attribute	Format
1	time_response	Domain lookup time response
2	domain_spf	Domain has SPF 2
3	asn_ip	ASN 3
4	time_domain_activation	Domain activation time (in days)
5	time_domain_expiration	Domain expiration time (in days)
6	qty_ip_resolved	Number of resolved IPs
8	qty_nameservers	Number of resolved NS4
9	qty_mx_servers	Number of MX 5servers
10	ttl_hostname	Time-To-Live (TTL)
11	tls_ssl_certificate	Has valid TLS 6/SSL 7certificate
12	qty_redirects	Number of redirects
13	url_google_index	Is URL indexed on Google
14	domain_google_index	Is domain indexed on Google
15	url_shortened	Is URL shortened
16	**phishing**	**Is phishing website**

B. Classifiers

This work used the above data set to compare the performance of two classifiers: The Decision Tree classifier and Naive Bayesian classifier.1) Decision tree: Decision trees are non-parametric classifiers. As its name indicates, a decision tree is a tree structure, where each non-terminal node denotes a test on an attribute, each branch represents an outcome of the test, and the leaf nodes denote classes. The basic algorithm for decision tree induction is a greedy algorithm that constructs the decision tree in top-down recursive divide-and-conquer manner [12]. At each non-terminal node, one of attributes is chosen for the split. The attribute that gives the maximum information gain is chosen for the split. A well-known algorithm for decision trees is the C4.5 algorithm where entropy is used as a criterion to calculate the information gain. The information gain is defined as the difference between the entropy before the split and the entropy after the split. Equations to calculate information gain are below.

$$\mathbf{H\ (T) = - \sum p_i \log_2 (p_j)}$$

$$\mathbf{Hs\ (T) = - \sum p_i Hs\ (T_i)}$$

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 7, July 2021

DOI 10.17148/IJARCCE.2021.10734

Gain (S)= H(T) – Hs(T)

Where **H (T)** the entropy before the split, **Hs (T)** is the entropy after the split, and p_i is probability of class j. One of the main concerns with the decision tree classifier is that it over fits the training data.

2) Naïve bayes' classifier: This classifier calculates the posterior probability for each class and assigns the sample to the class with the maximum probability [13]. The posterior probability for class i is given by Equation (2) and can be calculated from the training set data.

$P (C_i / x) = P (x / C_i) P(C_i)$

where $P (x/C_i) = \prod P (x_k / C_i)$

IV. RESULTS AND CONCLUSIONS

A. Results:

The data set used in this paper was downloaded from the results of the paper 'Datasets for phishing websites detection ' by **GregaVrbančič, IztokFisterJr. and ViliPodgorelec.[11].** The dataset in total features 111 attributes excluding the target phishing attribute, which denotes whether the particular instance is legitimate (value 0) or phishing (value 1). It consist of 111 features of 88647 sites that are of known class. The pareto principle (80:20) is used to separate the training and testing dataset. We used around 64000 rows for training and 24647 rows were used for testing.80 percent of the sample is used for training and the remaining 20 percent is used for testing. We got a astonishing result of accuracy of 0.95 for Decision Tree classifier and 0.84 for Naives bayes classifier.

B. Conclusions

In this work we implemented two classifiers using python scripts in Anaconda IDE which are decision tree and Naives Bayes' classifier. It is clear that decision tree outperformed the Naives Bayes' classifier. This is because Navies Bayes' classifier is good for datasets with multiple classifications whereas our system has only two classes. Decision trees classifier is a better choice because it is good for datasets with fewer classes and we had the least number possible (two classes).

V. FUTURE WORK

This research paper here has some limitations and it can be extended further. The dataset is fixed, the dataset doesn't give us the URLs instead it gives us the extracted features. So extra features cannot be added. Also some more classifiers can be used so that a better accuracy may be obtained in those techniques. Also DOM based classification can be added so that the system may not be only dependent on the URL but also the design of the webpage.

REFERENCES

[1] N. Lord, "How to Recognize and Avoid Phishing Scams". https://www.consumer.ftc.gov/articles/how-recognize-and-avoid-phishing-scams, 2018.

[2] N. Sadeh, A. Tomasic, and I Fette, "Learning to detect phishing emails", Proceedings ofthe16thinternational conference on world wide web, pp.649–656, 2007.

[3] J. Ma, S. S. Savag, G. M. Voelker, "Learning to detect malicious URLs", ACM Transactions on Intelligent Systems and technology, vol. 2, no. 9, pp 30:1-30:24, 2011.

[4] S. Purkait, "Phishing counter measures and their effectiveness–literature review", Information Management & Computer Security, vol. 20, no. 5, pp. 382–420, 2012.

[5] N. Abdelhamid, A. Ayesh, F. Thabtah, "Phishing Detection based Associative Classification", Data Mining. Expert Systems with Applications (ESWA), vol. 41, pp 5948-5959, 2014.

[6] D. R. Patil and J. Patil, J., "Survey on malicious web pages detection techniques", International Journal of u-and e-Service, Science and Technology, vol. 8, no. 5, pp. 195–206, 2015.

[7] W. Hadi, F. Aburrub, and S, Alhawari, "A new fast associative classification algorithm for detecting phishing websites", Applied Soft Computing vol. 48, pp 729-734, 2016.

[8] R. K. Nepali and Y. Wang, Y., "You look suspicious!! Leveraging visible attributes to classify malicious short urls on twitter", 2016 49th Hawaii International Conference on System Sciences (HICSS). IEEE, pp. 2648–2655, 2016.

[9] M. Kuyama, Y. Kakizaki, and R. Sasaki, "Method for detecting a malicious domain by using whois and dns features", The Third International Conference on Digital Security and Forensics (DigitalSec2016), p. 74, 2016.

[10] D. Sahoo, C. Liu, and C. H. Hoi, "Malicious URL detection using machine learning: A Survey", https://arxiv.org/abs/1701.07179, 2017.

[11] Datasets for phishing websites detection by GregaVrbančič, IztokFisterJr. and ViliPodgorelec https://doi.org/10.1016/j.dib.2020.106438

[12] J. R. Quinlan, "Induction of Decision Trees", Machine Learning, vol. 1, no. 1, pp. 81–106, 1986.

[13] J. Han and M. Kamber, "Data Mining–Concepts and Techniques", Morgan Kaufmann, San Francisco, CA, 2011.

[14] V. N. Vapnik, "Support-vector networks", Machine Learning, vol. 20 no. 3, pp 273–297, 1995.