ISSN 2616-5775 Vol. 6, Issue 5: 91-94, DOI: 10.25236/AJCIS.2023.060512

A phishing website detection system based on machine learning methods

Chengge Duan[1,a,*], Minze Wang[2,b], Xin Lu[2,c], Junming Wang[3,d]

[1]*Suzhou Public Security Bureau, Suzhou, Jiangsu, China*
[2]*School of Social Computing, Xi'an Jiaotong-Liverpool University, Suzhou, Jiangsu, China*
[3]*The Third Research Institute of Ministry of Public Security, Shanghai, China*
[a]*secrecyge@126.com*, [b]*Minze.Wang22@student.xjtlu.edu.cn*, [c]*Xin.Lu22@student.xjtlu.edu.cn*,
[d]*18265379@qq.com*
[*]*Corresponding author*

Abstract: *In today's Internet age, phishing attacks are a common means of cyberattacks. Most existing URL-based anti-phishing technologies are simple and effective, but lagging, while machine learning and deep learning-based approaches can effectively improve detection efficiency. This study advocates the use of TF-IDF for website data preprocessing followed by a random forest model to achieve phishing website feature classification. The final experimental results show that the model accuracy of the random forest algorithm based on machine learning to judge phishing websites is high and the anti-phishing capability is superior.*

Keywords: *Phishing Website Detection; Random Forest Model; TF-IDF; Machine Learning*

1. Introduction

Since the 21st century, computer Internet technology has been rapidly developed and applied, and with the continuous emergence of various new services and applications on the Internet, people have become more and more dependent on the Internet. The rapid development of the Internet industry has been accompanied by the registration of a large number of domain names, a spurt of new Web sites, and the constant updating of Web pages. The phishing attack is a deceptive message issued to users through the Web site as an information carrier to lure Internet users to visit their fictitious phishing sites and leak sensitive information. Due to the Internet's open and shared nature, phishing attacks are spreading worldwide and becoming a global issue.

Detecting phishing website attacks is a long-term test for current domestic and international network security technologies, and the main phishing detection methods are currently divided into the following categories [1]: (1) black-and-white list detection techniques based on monitoring whether the URLs visited by browsers are in the list; (2) heuristic algorithm-based detection techniques, which is a method of training by extracting UPL features, structural features, and text content features [2] to obtain result predictions; (3) visual similarity-based detection techniques, which perform matching detection by extracting visual features of web page screenshots [3]; in the study of anti-phishing website techniques, we found that using machine learning methods for phishing website detection can effectively use new features that are constantly discovered and extracted to deal with the constantly refurbished phishing web pages. Research has shown that virous traditional machine learning algorithms such as decision trees, support vector machines (SVMs), and neural networks can be effectively used for phishing website detection [4]. However, if all the data features are fed into the classifier at once, it is likely to lead to a "dimensional disaster" [5] and reduce the recognition accuracy.

The Classification and Regression Tree (CART) [6] is used as a meta-classifier in the Random Forest (RF) technique, which was developed by Breiman [7] in 2001. The meta-classifier CART is built based on a few features that are far fewer than the number of all features, which can avoid the above-mentioned "dimensional catastrophe" problem. This paper proposes a random forest algorithm-based classifier model for detecting phishing websites after TF-IDF feature extraction [8] data preprocessing.

ISSN 2616-5775 Vol. 6, Issue 5: 91-94, DOI: 10.25236/AJCIS.2023.060512

2. Materials and Method

2.1 TF-IDF

The Term Frequency-Inverse Document Frequency (TF-IDF) is a weighted technique that is commonly used in information retrieval and data mining. It is often utilized to extract keywords from articles, and its algorithm is both simple and efficient, making it a popular choice for initial text data cleaning in industry settings. The frequency of a given word appearing in a document affects the TF-IDF value for a given input text. TF represents the number of times a given word appears in a document, while IDF represents the logarithm of the total number of documents divided by the number of documents that contain the word. Higher TF-IDF values indicate that a word occurs more frequently in a given text, while simultaneously appearing less frequently in other articles. The formulas for calculating TF and IDF values are provided below:

$$TF_{ij} = \frac{n_{i,j}}{\sum_k n_{k,j}} \tag{1}$$

$$IDF_i = \log \frac{|D|}{1+|\{j:T_i \in D_j\}|} \tag{2}$$

In the context of the TF-IDF algorithm, $n_{i,j}$ represents the number of times a given word T_i appears in a document D_j, $\sum_k n_{k,j}$ represents the total number of occurrences of all words in the document D_j, $|D|$ represents the total number of documents in the corpus, and $|\{j: T_i \in D_j\}|$ represents the number of documents containing the word of T_i. If the word does not appear in any document, then $|\{j: T_i \in D_j\}|$ is equal to zero. Due to its simple implementation and high computational efficiency, the TF-IDF algorithm has been applied in the pre-processing step of HTML text for phishing websites.

2.2 Random Forest

The Random Forest (RF) algorithm, proposed by Leo Breiman and Adele Cutler, employs multiple weak decision trees to predict and classify samples, and then takes a vote on the classification results of each subtree to obtain a classifier equivalent to a strong decision tree. The core idea of RF is to offset the classification errors of a large number of weak classifiers against each other, resulting in an accurate and comprehensive decision-making result, thereby achieving a higher classification efficiency than that of a single complex classification algorithm. RF is a multi-decision tree classifier that requires a training set with replacement to construct multiple simple classifiers. The sample size for each with-replacement sampling is the same as that of the training set. Then, each sub-dataset randomly selects classification features to construct a sub-decision tree, outputting a classification result obtained based on the feature set as the classification boundary. Figure 1 shows the principle diagram for the RF algorithm.

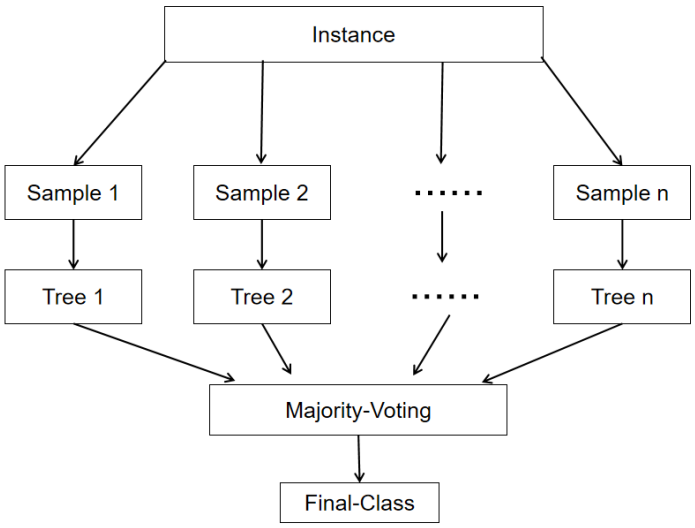

Figure 1: A Schematic diagram of the RF classification algorithm.

ISSN 2616-5775 Vol. 6, Issue 5: 91-94, DOI: 10.25236/AJCIS.2023.060512

The Random Forest algorithm has the following three main characteristics: firstly, the selection of training samples is random; secondly, the selection of feature subsets is random; thirdly, all tree models grow freely without pruning. These characteristics can be represented by Equation 2:

$$f = \{h(X, \theta_k), k = 1,2, \dots\} \tag{3}$$

In the equation, $h(X, \theta_k)$ represents a classification and regression tree (CART) constructed using the CART algorithm, and θ_k is an independently and identically distributed random vector. The Random Forest algorithm first uses the Bagging method to generate subsets of training samples with differences. Then it uses the random subspace method to select some attributes to construct multiple classification and regression trees using the CART algorithm.

The Bagging algorithm randomly selects n samples from an initial training set with a total sample size of n to form a new sample set. The probability that a single sample in the initial training set is not selected is $(1 - 1/n)^n$. As n approaches infinity:

$$\lim_{n\to\infty} (1 - 1/n)^n \approx 0.368 \tag{4}$$

Therefore, this paper deduces that approximately 36.8% of the samples in the initial sample set will not be present in the new training sample set, thus ensuring the diversity of the training sample set. The CART algorithm utilizes the Gini index as the splitting criterion, but the CART decision tree in the Random Forest algorithm grows freely without any pruning operations, ensuring the randomness of the decision tree and mitigating the overfitting phenomenon.

The primary advantage of the Random Forest algorithm is that although each tree is independent and prone to errors, randomly extracting a large amount of data yields decision trees with low errors and non-correlation. Eventually, only the classification trees with accurate classification are retained. This approach aligns with the bagging idea of the Random Forest algorithm. Once the influence of multiple weak classifiers has been voted and offset, an unbiased decision of a strong classifier is obtained.

3. Experiments and Analysis

3.1 Data set

In this study, data was obtained from public datasets on the internet. The blacklist contained over 30,000 entries, while the whitelist contained over 20,000 entries. For the analysis, 25 entries were randomly selected from both the black and white lists, resulting in a total of 50 entries used to evaluate the accuracy of the machine learning-based random forest algorithm and TF-IDF algorithm applied in this experiment. Regarding model training, Scikit-Learn and Numpy were utilized for machine learning model training and statistical data analysis. Scikit-Learn is advantageous due to its diverse models, comprehensive functions, and rich data resources. It can implement various supervised or unsupervised learning models, and is simple, fast, and efficient to use. Numpy is a Python language extension library based on Scikit-Learn, which supports a large number of multidimensional array and matrix operations, and provides a wealth of mathematical function libraries. In this experiment, the legitimate and phishing websites were trained separately from the training dataset, with Scikit-Learn classifying the URLs and identifying the specific features of legitimate and illegitimate websites. These string features were then passed to the TF-IDF algorithm for computing TF-IDF scores. Numpy was used to import the dataset and returned the progress to the user interface to notify the training process. The Tokenizer library was used for feature word extraction from the URL data, which divided each URL into independent tokens, specifically using the bytelevelBPETokenizer method in Tokenizer. After the feature classification and model training, the Joblib library was used to save the model to disk, allowing the model to be re-run at any time.

This experiment was conducted using Windows11 64-bit operating system, Python 3.9, Anaconda 3, VS Code, and Spyder.

3.2 Experimental results and comparative analysis

This experiment yielded four types of results: model true phishing (MTP) and model false phishing (MFP) predicted as phishing websites, and model true not phishing (MTN) and model false not phishing (MFN) predicted as legitimate web pages. We also defined three metrics for evaluation:

ISSN 2616-5775 Vol. 6, Issue 5: 91-94, DOI: 10.25236/AJCIS.2023.060512

accuracy (A), precision (P) and recall (R).

The corresponding formulas are as follows:

$$A = \frac{MTP+MTN}{MTP+MFP+MTN+MFN} \tag{5}$$

$$P = \frac{MTP}{MTP+MFP} \tag{6}$$

$$R = \frac{MTP}{MTP+MFN} \tag{7}$$

After using model to test phishing websites and legitimate web pages, the following data were obtained:

Table 1: Model Test Results Table

Data Types	Results
MTP	20
MFP	5
P-ALL	25
MTN	22
MFN	3
N-ALL	25

The results of this test can be obtained after bringing the corresponding data into the formula, with accuracy (*A*) at 82.00%, precision (*P*) at 80.00%, and recall (*R*) at 86.96%. Based on the overall results, the machine learning-based random forest algorithm for detecting phishing websites demonstrated high accuracy, and both precision and recall performed exceptionally well.

4. Conclusions

To further improve the training accuracy and reduce the training time of the model, logistic regression can be considered as a feature classifier, and the HTML and URL features can be fused into a feature vector to reduce training time and improve training efficiency. In addition to traditional methods of identifying phishing websites using web page URLs, image recognition frameworks can be utilized to extract image features of phishing websites or illegal website cookies can be called upon to assist in identifying phishing websites. Finally, combined with the deep learning frameworks, the accuracy, precision, and recall of phishing website detection can be improved.

Acknowledgement

This work is supported by Key Lab of Information Network Security of Ministry of Public Security (The Third Research Institute of Ministry of Public Security), ID:C21613.

References

[1] Kim Y G, Cho S, Lee J S, et al. Method for Evaluating the Security Risk of a Website Against Phishing Attacks.[J]. Lecture Notes in Computer Science, 2008, 5075:21-31.
[2] Zhang Y, Hong J I, Cranor L F. Cantina: a content-based approach to detecting phishing web sites [C]// Proceedings of the 16th international conference on World Wide Web. ACM, 2007:639-648.
[3] Liu W, Deng X, Huang G, et al. An antiphishing strategy based on visual similarity assessment [J]. IEEE Internet Computing, 2006, 10(2): 58-65.
[4] Xia T, Chai Y, & Wang T. Improving SVM on web content classification by document formulation [C] //2012 7th International Conference on Computer Science & Education (ICCSE), 2012: 110-113.
[5] Breiman L. Statistical Modeling: The Two Cultures (with comments and a rejoinder by the author). Statistical Science, 2001, 16, 199-231.
[6] Breiman L, Friedman, J, Olshen R, & Stone C. Classification and Regression Trees [M]. Boca Raton, FL: CRC Press, 1984:18-58.
[7] Breiman L. Random Forests--random features [J]. Machine Learning, 1999, 45(1):5-32.
[8] Manjari K, Rousha S, et al. Extractive Text Summarization from Web pages using Selenium and TF-IDF algorithm[C]//2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184), 2020: 648-652.