PHISHING WEBSITE DETECTION

MSC 105 Data Mining
Department of Computer Science
University of Delhi

Jagriti Mittal (22)
Khushi Jain (25)
Yashi Sharma (58)

Equal contribution from each member

Problem

Cybercriminals employ deceptive tactics, often exploiting human vulnerability, to gain unauthorized access to sensitive information.

Phishing attacks have evolved from simple email scams to highly targeted and convincing campaigns that can compromise personal and organizational security. The consequences of falling victim to such attacks can be severe, ranging from financial losses to data breaches with far-reaching implications.

Phishing, at its core, is a form of cybercrime that employs deceptive tactics to trick individuals into revealing sensitive information such as usernames, passwords, and financial details. It's a term that encapsulates various techniques, often exploiting human psychology and trust to gain unauthorized access.

Common Tactics Used by Phishers:

- Email Spoofing
- Social Engineering
- Phishing Websites

In this project, we have tried to solve the problem of phishing websites using machine learning models.

Related work

Research Paper 1

A new hybrid ensemble feature selection framework for machine learning-based phishing detection system By Kang Leng Chiewa, Choon Lin Tan, KokSheik Wong, Kelvin S.C. Yong, Wei King Tiong

This research paper proposes a new feature selection framework for machine learning-based phishing detection systems called the **Hybrid Ensemble Feature Selection (HEFS)**.

The framework consists of two phases: the first phase utilizes the Cumulative Distribution Function gradient **(CDF-g) algorithm** to generate primary feature subsets. In contrast, the second phase derives baseline features from the secondary feature subsets. Experimental results show that HEFS, when integrated with the Random Forest classifier, performs best, identifying baseline features that correctly distinguish **94.6% of phishing** and legitimate websites using only 20.8% of the original features.

Authors selected **5000 phishing webpages** based on URLs from PhishTank2 and OpenPhish3, and another **5000 legitimate webpages** based on URLs from Alexa4 and the Common Crawl5 archive.

The researchers utilized filter measures, such as **Information Gain, Relief-F, and Symmetrical Uncertainty**, to rank features and generate primary feature subsets. The CDF-g algorithm was then applied to identify the optimal cut-off rank of features. The hybrid ensemble strategy combined **data perturbation and function perturbation** techniques to produce secondary feature subsets and the final baseline feature set.

The HEFS framework is a highly desirable and practical feature selection technique for machine learning-based phishing detection systems. It outperforms existing techniques in terms of accuracy and efficiency, and the derived baseline features are effective in distinguishing between phishing and legitimate websites. The framework can be universally applied to different datasets and provides a benchmark for evaluating future phishing detection techniques.

Advantages of this methodology

1. Enhances the detection accuracy and computational efficiency.
2. It is not affected by the order of the instances present in the dataset as it randomly partitions the datasets into parts and uses data permutation and feature permutation for feature extraction.

Disadvantages of this methodology

1. Cumulative distribution frequency gradient uses gradient for calculating the cut-off rank, therefore it can have limitations like, low convergence rate, unbalanced selection effect, and biased estimation.
2. The authors had some interesting findings that suggest features such as NumDots, UrlLength, AtSymbol, NoHttps, and IpAddress which are considered essential for phishing detection, actually only contribute little to the accuracy of a phishing detection technique.

Research Paper 2

An effective detection approach for phishing websites using URL and HTML features By Ali Aljofey, Qingshan Jiang, Abdur Rasool, Hui Chen, Wenyin Liu, Qiang Qu & YangWang

This paper provides an efficient solution for phishing detection that extracts the features from a website's URL and HTML source code. Specifically, we proposed a hybrid feature set including URL character sequence features, various hyperlink information, plaintext, and noisy HTML data-based features within the HTML source code. These features are then used to create a feature vector required for training the proposed approach by the XGBoost classifier.

The dataset consists of 60,252 webpages and their HTML source codes, wherein 27,280 are phishing and 32,972 ones are benign

Their approach extracts and analyzes different features of suspected web pages for effective identification of large-scale phishing offenses. The main contribution of this paper is the combined uses of these feature sets.

To improve the detection accuracy of phishing web pages, the authors have proposed eight new features. The proposed features determine the relationship between the URL of the webpage and the webpage content.

From the experimental results, it is noticed that TF-IDF character level features outperformed other features with significant accuracy, precision, F-Score, Recall, and AUC using the XGBoost classifier. Hence, we implemented the TF-IDF character level technique to generate text features (F2) of the webpage.

The given approach detects phishing websites based on URL and HTML features. XGBoost Classifier with all types of features gives the best result with an accuracy of 96.76%. As blockchain technology emerges, it becomes a perfect target for phishing attacks.

Advantages of this methodology

1. It is a machine learning based detection method and hence can be used to classify any url. This makes this approach dynamic and more suitable.

2. It classifies the websites based on both url and web page content, this enhances its accuracy.

Disadvantages of this methodology

1. Since it also considers HTML source code for classification, therefore it is web-page language dependent.
2. It does not identify the textual content hidden in embedded objects.
3. Large training time (due to high dimensional vector)
4. Not sufficiently capable of detecting attached malware

Methodology

Dataset used :-

Our dataset is a robust collection comprising 11,055 instances, each associated with 32 features. These features are intricately crafted to encapsulate various aspects of a website's URL and its associated characteristics.

The richness of our dataset lies in its ability to capture the nuanced details that distinguish legitimate websites from potential phishing threats. These features are derived from a meticulous analysis of the URL structure and other attributes associated with each website.

Implementation

1. Data preprocessing - Checked the dataset for the presence of null and missing values and dropped the id column, due to its unique value and non predictive nature.
   Checked the class distribution of the target class in the dataset.
2. Feature Selection - Used the following measures to select the features
   a) Pearson correlation

1. Calculated the correlation between features, and dropped the one of the features from the pair of features that had correlation more than 70%.
2. Calculated the correlation between features and result and dropped the features that had less than 0.01 correlation with the result.

b) Information Gain - Calculated mutual information gain between features and target class column and dropped the columns with zero mutual information gain.

c) Variance - Calculated variance of features, to identify any constant features in the dataset.Dropped the features with zero variance.

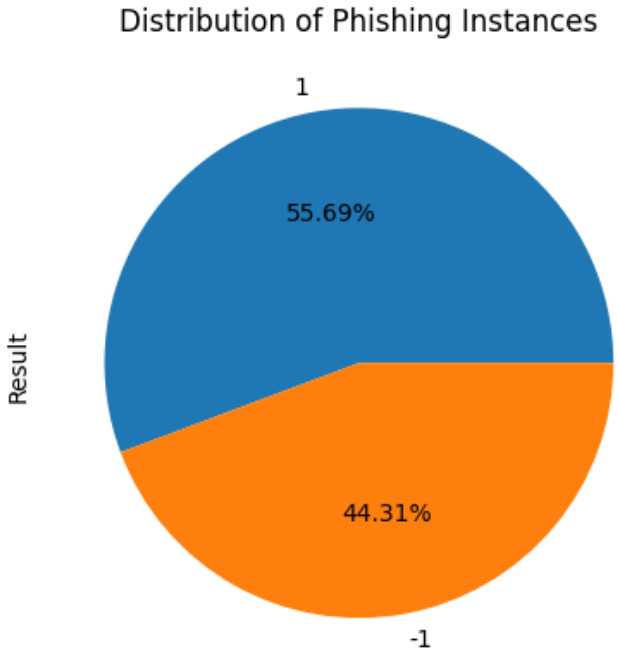
3. Classification Models - After feature selection, splitted the data into test and train data. 40% of the data was taken as test data and the remaining 60% of the data was taken as training data. Applied the following classification models and calculated their accuracy, precision, recall score, f score, and other accuracy measures on the selected features:-
   a) Decision Tree - With entropy as the measure for classification and limiting the maximum depth of the decision tree to 10 levels.
   b) Naive Bayes
   c) Random Forest - Limiting the maximum depth of the tree to 10 levels.
   d) XGBoost Classifier - With a learning rate of 0.01 and limiting the maximum depth to 4 levels.

Experimental Results and Discussion

By performing the above mentioned experiments following results were observed : -

1. By Data processing on one feature, a unique id column was dropped as the dataset does not have any missing or null values.

This pie chart illustrates the distribution of the target class in the dataset. Here 1 indicates legitimate websites and -1 indicates phishing websites.
Therefore as depicted in the pie chart dataset contains 55.69% of legitimate website instances and around 44.31% of phishing websites.



Source of the chart: From the experiments performed in code

Since the data is roughly equally distributed among classes, therefore stratified sampling was not performed.

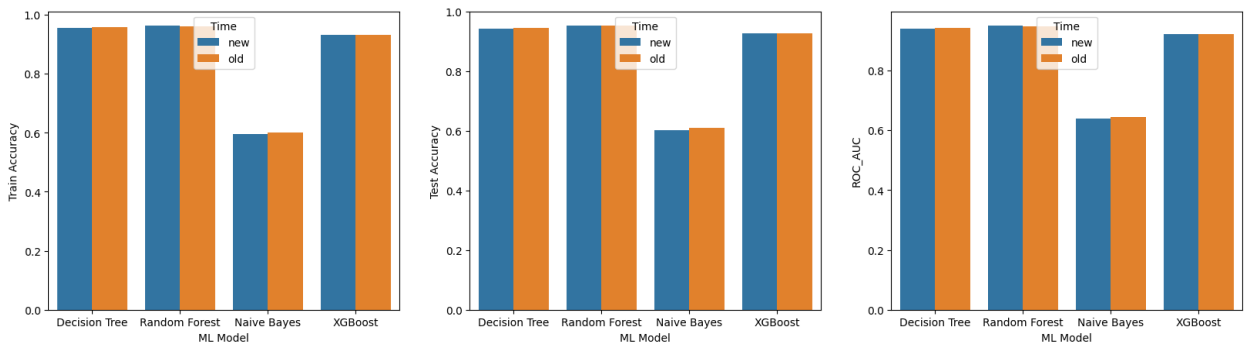
2. By feature selection, 12 features were dropped, and 19 features were retained: -
   - 7 features were highly correlated with other features
   - 2 features were very less correlated with the result
   - 3 features had 0 information gain with the result
   - The dataset had no constant feature so no feature was dropped using variance.
3. Various classification models were applied to the selected features and their training accuracy, test accuracy, precision, recall, f1 score, and roc_accuracy were calculated and stored in a table for comparison.

The following table depicts the accuracy of various models after feature selection.

	ML Model	Train Accuracy	Test Accuracy	Precision	Recall	f1 score	ROC_AUC	Time
0	Decision Tree	0.9566	0.9412	0.9436	0.9498	0.9467	0.9403	new
1	Random Forest	0.9631	0.9525	0.9408	0.9749	0.9576	0.9500	new
2	Naive Bayes	0.5945	0.6027	1.0000	0.2773	0.4341	0.6386	new
3	XGBoost	0.9308	0.9263	0.9068	0.9650	0.9350	0.9220	new

Source of the table : From the experiments performed in code

The following bar chart depicts the test accuracy, train accuracy, and roc_accuracy of classification models before feature selection and after feature selection.



Source of the chart : From the experiments performed in code

Here the time legend symbolizes the results obtained on applying classification models after feature selection and old symbolizes the results obtained on applying classification without feature selection.

Conclusion

Based on the results of the experiments performed we can conclude that **Random Forest Classifier gave the best results** with the train accuracy of 0.96 and test accuracy of 0.95. In addition to it, we can also conclude

that the accuracy of the classification was increased after the feature selection, even if it is increased in a very small proportion, but it has increased.

Limitations of the implementation done

1. For now, from a given URL it cannot predict whether a given URL is a phishing website or not, without giving the parameter's value. Thus it is considering parameters as hard-coded and not performing feature extraction from the URL.
2. This methodology works on URL features and does not consider any textual content of the websites.

Future Work

The shortcomings of the implementation can be overcome by:-

1. Including textual features along with url features for classification.
2. Including feature extraction code, so that features can be extracted from a given URL.
3. Other classification algorithms can be used to compare the results and conclude the best classification algorithm.

References

1. A new hybrid ensemble feature selection framework for machine learning-based phishing detection system By Kang Leng Chiewa, Choon Lin Tan, KokSheik Wong, Kelvin S.C. Yong, Wei King Tiong
2. An effective detection approach for phishing websites using URL and HTML features By Ali Aljofey, Qingshan Jiang, Abdur Rasool, Hui Chen, Wenyin Liu, Qiang Qu & YangWang
3. Detection of Phishing Websites using Machine Learning By Atharva Deshpande, Omkar Pedamkar, Nachiket Chaudhary, Dr. Swapna Borde
4. https://towardsdatascience.com/tf-idf-for-document-ranking-from-scratch-in-python-on-real-world-dataset-796d339a4089

5. Detection Of Phishing Websites Using Data Mining By Mr.Aniket Kote, Mr.Sanket Kharche, Mr.Pravin Aware, Mr.Abhishek pangavhane
6. A Hybrid Model to Detect Phishing-Sites using Supervised Learning Algorithms By M. Amaad Ul Haq Tahir , Sohail Asghar , Ayesha Zafar, Saira Gillani