**Detection-of-phishing-websites-from-URLs-using-IBM-Watson-studio**

**Group Members:-**

Aditya Kunal Bhate - 19BCE10046

Shlok Sharma - 19BCE10102

Nidhi Gupta - 19BCE10423

Srushti Prashant Shintre - 19BCE100432

**INTRODUCTION**

Detecting phishing websites using machine learning

Phishing is a type of attack where an attacker tricks the victim to give up sensitive information such as login credentials by disguising as a trustworthy entity. In this application, we will try to detect a phishing website using the features that differentiate these domains from legitimate ones. We will create our own dataset, train and test various machine learning models using Jupyter Notebooks on IBM Watson studio and deploy the best model to be used by the application for detection.

Various features that are used to create the dataset are as follows:

1. Using IP Address - check if the URL has an IP address in it
2. HTTPS - checking the existence of 'https', trusted certificate authority, and age of certificate
3. URL Short - check if the URL has been shortened
4. Having @ symbol - it leads the browser to ignore everything preceding the '@' symbol
5. Having a double-slash - means that the user will be redirected ([http://www.legitimate.com//http://www.phishing.com](http://www.legitimate.com/http:/www.phishing.com))
6. Domain registration Length - Trustworthy domains are regularly paid for several years
7. favicon - favicon loaded from the domain or not
8. Existencence of https token in the domain part of the URL
9. Request URL - examines whether the external objects contained within a webpage are loaded from another domain
10. URL of Anchor - If the tags and the website have different domain names
11. Links in tags - It is expected that tags (Meta, Script, and Link) are linked to the same domain of the webpage.
12. Server Form Handler - If it is blank or contains any other domain name
13. Submitting information to the email
14. Abnormal URL - if the domain name (from whois) is not in URL
15. redirect count
16. invisible iframe
17. Age of domain
18. web traffic - google rank for the page
19. statistical report - match it with top 10 domains and top 10 IPs from PhishTank

Getting started

1. Sign up for an [IBM Cloud account](https://console.bluemix.net/registration/)
2. Login to the [IBM Watson Studio](https://www.ibm.com/cloud/watson-studio)
3. Install Python3.7
4. Install dependencies

pip install -r packages.txt

Creating Dataset

The dataset created for this application uses around 250 legitimate and 250 phishing urls with 20 features each as mentioned above. You can add more data and features (feature\_extraction.py) to the project to create your own dataset as shown below.

The URLs for phishing websites were retrieved from [here](https://www.phishtank.com/developer_info.php) (verified\_online.csv) and The URLs for legitimate websites were retrieved from [here](http://s3.amazonaws.com/alexa-static/top-1m.csv.zip) (top1m.csv)

1. Create the dataset for the phishing websites

python create\_dataset.py <file\_with\_phishing\_url> <number\_of\_urls\_to\_use> <output\_file> <target\_value>

python create\_dataset.py verified\_online.csv 500 dataset2.csv 1

1. Create the dataset for the legitimate websites

python create\_dataset.py <file\_with\_legitimate\_url> <number\_of\_urls\_to\_use> <output\_file> <target\_value>

python create\_dataset.py top1m.csv 500 dataset2.csv 0

Preprocess the data, build machine learning models and test them

1. Sign up for Watson Studio

Sign up for IBM's [Watson Studio](https://dataplatform.ibm.com/).

2. Create a new Project

Note: By creating a project in Watson Studio a free tier Object Storage service will be created in your IBM Cloud account. Take note of your service names as you will need to select them in the following steps.

* On Watson Studio's Welcome Page select New Project.
* Choose the Data Science option and click Create Project.
* Name your project, select the Cloud Object Storage service instance and click Create

3. Upload the dataset

* Drag and drop the dataset (csv) file you just created to Watson Studio's dashboard to upload it to Cloud Object Storage.

4. Import notebook to Watson Studio

* Create a New Notebook.
* Import the notebook found in this repository
* Give a name to the notebook and select a Python 3.5 runtime environment, then click Create.

5. Import the dataset into the notebook

To make the dataset available in the notebook, we need to refer to where it lives. Watson Studio automatically generates a connection to your Cloud Object Storage instance and gives access to your data.

* Go to the Files section to the right of the notebook and click Insert to code for the data you have uploaded. Choose Insert pandas DataFrame.

6. Follow the steps in the notebook

The steps should allow you to understand the dataset, analyze and visualize it. You will then go through the preprocessing and feature engineering processes to make the data suitable for modeling. Finally, you will build some machine learning models and test them to compare their performances.

7. Deploy the best model on IBM Cloud

1. Navigate to your project and add a new machine learning model.
2. Give it a name, choose a machine learning service, select model builder as model type as logistic regression is one of the best model for our dataset and is available in the builder, select the default runtime and select Manual.
3. Add the reduced dataset to the model.
4. Add a deployment
5. Get the deployment url and the machine learning model instance tokens.
6. Replace the deployment url and tokens in the check\_url.py file

Test a URL

1. python check\_url.py <url>
2. Nowadays Phishing becomes the main area of concern for security researchers because it is not difficult to create a fake website that looks so close to a legitimate website. Experts can identify fake websites but not all the users can identify the fake website and such users become the victim of the phishing attack. The main aim of the attacker is to steal bank account credentials. Phishing attacks are becoming successful because of a lack of user awareness. Since phishing attack exploits the weaknesses found in users, it is very difficult to mitigate them but it is very important to enhance phishing detection techniques. Phishing may be a style of broad extortion that happens once a pernicious web site act sort of a real one memory that the last word objective to accumulate unstable info, as an example, passwords, account focal points, or MasterCard numbers. all the same, this means that there square measure some of contrary to phishing programming
3. and techniques for recognizing potential phishing tries in messages and characteristic phishing substances on locales, phishes think about new and crossbreed procedures to bypass the open programming and frameworks. Phishing may be a fraud framework that uses a mixture of social designing what is additional, advancement to sensitive and personal data, as an example, passwords associate degree open-end credit unpretentious elements by presumptuous the highlights of a reliable individual or business in electronic correspondence. Phishing makes use of parody messages that square measure created to seem substantial and instructed to start out from true blue sources like money-connected institutions, online business goals, etc, to draw in customers to go to phony destinations through joins given within the phishing websites.

**STATE OF THE ART (LITERATURE SURVEY)**

1. H. Huang et al., (2009) proposed the frameworks that distinguish the phishing utilizing page section similitude that breaks down universal resource locator tokens to create forecast preciseness phishing pages normally keep its CSS vogue like their objective pages.
2. S. Marchal et al., (2017) proposed this technique to differentiate Phishing websites depends on the examination of authentic site server log knowledge. An application Off-the- Hook application or identification of a phishing website. Free, displays a couple of outstanding properties together with high preciseness, whole autonomy, nice language freedom, speed of selection, flexibility to dynamic phish, and flexibility to advancement in phishing ways.
3. Mustafa Aydin et al. proposed a classification algorithm for phishing website detection by extracting websites' URL features and analyzing subset-based feature selection methods. It implements feature extraction and selection methods for the detection of phishing websites. The extracted features about the URL of the pages and composed feature matrix are categorized into five different analyses as Alphanumeric Character Analysis, Keyword Analysis, Security Analysis, Domain Identity Analysis, and Rank Based Analysis. Most of these features are the textual properties of the URL itself and others are based on third parties services.
4. Samuel Marchal et al. presents PhishStorm, an automated phishing detection system that can analyze in real time any URL in order to identify potential phishing sites. Phish storm is proposed as an automated real-time URL phishingness rating system to protect users against phishing content. PhishStorm provides phishingness score for URL and can act as a Website reputation rating system.
5. Fadi Thabtah et al. experimentally compared large numbers of ML techniques on real phishing datasets and with respect to different metrics. The purpose of the comparison is to reveal the advantages and disadvantages of ML predictive models and to show their actual performance when it comes to phishing attacks. The experimental results show that Covering approach models are more appropriate as anti- phishing solutions. Muhemmet Baykara et al. proposed an application which is known as Anti Phishing Simulator, it gives information about the detection problem of phishing and how to detect phishing emails. Spam emails are added to the database by Bayesian algorithm. Phishing attackers use JavaScript to place a legitimate URL of the URL onto the browsers address bar. The recommended approach in the study is to use the text of the e-mail as a keyword only to perform complex word processing.

**PROJECT DESCRIPTION**

1. We have developed our project using a website as a platform for all the users. This is an interactive and responsive website that will be used to detect whether a website is legitimate or phishing. This website is made using different web designing languages which include HTML, CSS, JavaScript and Django.
2. The basic structure of the website is made with the help of HTML. CSS is used to add effects to the website and make it more attractive and user-friendly. It must be noted that the website is created for all users, hence it must be easy to operate with and no user should face any difficulty while making its use. Every naive person must be able to use this website and avail maximum benefits from it.
3. The website shows information regarding the services provided by us. It also contains information regarding ill- practices occurring in today’s technological world. The website is created with an opinion such that people are not only able to distinguish between legitimate and fraudulent website but also become aware of the mal-practices occurring in the current world. They can stay away from the people trying to exploit one’s personal information, like email address, password, debit card numbers, credit card details, CVV, bank account numbers, and the list goes on.
4. The dataset consists of different features that are to be taken into consideration while determining a website URL as legitimate or phishing.
5. The components for detection and classification of phishing websites are as follows:
6. Address Bar based Features
7. Abnormal Based Features
8. HTML and JavaScript Based Features
9. Domain Based Features
10. Address Bar based Features
11. Using the IP address
12. If IP address is used instead of domain name in the URL
13. e.g. 125.98.3.123 the user can almost be sure someone is trying to steal his personal information.
14. Long URL to hide the Suspicious Part
15. Phishers can use long URL to hide the doubtful part in the address bar.
16. Using URL shortening services TinyURL
17. URL shortening is a method on the World Wide Web in which a URL may be made considerably smaller in length and still lead to the required webpage.
18. URLs having @ symbol
19. Using @ symbol in the URL leads the browser to ignore everything preceding the @ symbol and the real address often follows the @ symbol.
20. Redirecting using //
21. The existence of // within the URL path means that the user will be redirected to another website.
22. Adding Prefix or Suffix Separated by (-) to the Domain
23. The dash symbol is rarely used in legitimate URLs. Phishers tend to add prefixes or suffixes separated by (-) to the domain name so that users feel that they are dealing with a legitimate webpage.
24. Sub Domain and Multi Sub Domains
25. Let us assume we have the following link: http://www.hud.ac.uk/students/. A domain name might include the country-code top-level domains (ccTLD).
26. HTTPs (Hyper Text Transfer Protocol with Secure Sockets Layer)
27. The existence of HTTPS is very important in giving the impression of website legitimacy, but this is clearly not enough.
28. Domain Registration Length
29. Based on the fact that a phishing website lives for a short period of time, we believe that trustworthy domains are regularly paid for several years in advance. In our dataset, we find that the longest fraudulent domains have been used for one year only.
30. Favicon
31. A favicon is a graphic image (icon) associated with a specific webpage.
32. Using Non-Standard Port
33. This feature is useful in validating if a particular service is up or down on a specific server.
34. The existence of HTTPS Token in the Domain Part of the URL
35. The phishers may add the HTTPS token to the domain part of a URL in order to trick users.
36. Abnormal Based Features
37. Request URL
38. Request URL examines whether the external objects contained within a webpage such as images, videos and sounds are loaded from another domain.
39. URL of Anchor
40. An anchor is an element defined by the <a> tag. This feature is treated exactly as Request URL.
41. Links in <meta>, <Script> and <Link> tags
42. Given that our investigation covers all angles likely to be used in the webpage source code, we find that it is common for legitimate websites to use <Meta> tags to offer metadata about the HTML document; <Script> tags to create a client side script; and <Link> tags to retrieve other web resources.
43. It is expected that these tags are linked to the same domain of the webpage.
44. Server From Handler(SFH)
45. SFHs that contain an empty string or about:blank are considered doubtful because an action should be taken upon the submitted information.
46. Submitting Information to Email
47. Web form allows a user to submit his personal information that is directed to a server for processing. A phisher might redirect the users information to his personal email.
48. Abnormal URL
49. This feature can be extracted from WHOIS database. For a legitimate website, identity is typically part of its URL.
50. HTML and JavaScript Based Features
51. Website Forwarding
52. The fine line that distinguishes phishing websites from legitimate ones is how many times a website has been redirected. Status Bar Customization
53. Disabling Right Click
54. Phishers use JavaScript to disable the right-click function, so that users cannot view and save the webpage source code. This feature is treated exactly as Using onMouseOver to hide the Link.
55. Using Pop-Up Window
56. It is unusual to find a legitimate website asking users to submit their personal information through a pop-up window.
57. IFrame Redirection
58. IFrame is an HTML tag used to display an additional webpage into one that is currently shown.
59. Domain Based Features
60. Age of Domain
61. This feature can be extracted from WHOIS database. Most phishing websites live for a short period of time. By reviewing our dataset, we find that the minimum age of the legitimate domain is 6 months.
62. DNS Record
63. For phishing websites, either the claimed identity is not recognized by the WHOIS database or no records founded for the hostname. If the DNS record is empty or not found then the website is classified as Phishing, otherwise it is classified as Legitimate.
64. Website Traffic
65. This feature measuresthe popularity of the website by determining the number of visitors and the number of pages they visit.
66. Page Rank
67. PageRank is a value ranging from 0 to 1. PageRank aims to measure how important a webpage is on the Internet.
68. Google Index
69. This feature examines whether a website is in Googles index or not. When a site is indexed by Google, it is displayed on search results.
70. Number of Links Pointing to Page
71. The number of links pointing to the webpage indicates its legitimacy level, even if some links are of the same domain.
72. Statistical-Reports Based Feature
73. Fig.1. URL parts and features

**ALGORITHMS USED**

1. Two algorithms have been implemented to check whether a URL is legitimate or fraudulent.
2. Random forest algorithm creates the forest with number of decision trees. High number of tree gives high detection accuracy. Creation of trees is based on bootstrap method. In bootstrap method features and samples of dataset are randomly selected with replacement to construct single tree. Among randomly selected features, random forest algorithm will choose best splitter for classification.
3. Decision tree begins its work by choosing best splitter from the available attributes for classification which is considered as a root of the tree. Algorithm continues to build tree until it finds the leaf node. Decision tree creates training model which is used to predict target value or class in tree representation each internal node of the tree belongs to attribute and each leaf node of the tree belongs to class label.
4. Fig.2. Decision Tree Algorithm working
5. Fig.3. Random Forest Algorithm working

**PROJECT REQUIREMENTS**

***Hardware Requirements:-***

* 2GB RAM (minimum)
* 100GB HDD (minimum)
* Intel 1.66 GHz Processor Pentium 4 (minimum)
* Internet Connectivity
* Software Requirements:-
* WINDOWS 7 or higher
* Python 3.6.0 or higher
* Visual Studio Code
* Django
* HTML
* Dataset of Phishing Websites

**WORKING**

* We have collected unstructured data of URLs from Phishtank website, Kaggle website and Alexa website, etc.
* In pre-processing, feature generation is done where nin features are generated from unstructured data. These
* features are length of an URL, URL has HTTP, URL has suspicious character, prefix/suffix, number of dots, number of slashes, URL has phishing term, length of subdomain, URL contains IP address.
* After this, an organized dataset is made in which each detail incorporates the paired (0,1) which is then passed to the various classifiers.
* Next, we train the three unique classifiers and analyse their presentation based on exactness two classifiers utilized are Decision Tree and Random Forest algorithm.
* At that point, the classifier identifies the given URL dependent on the preparation information that is if the site is phishing it prompts the user that the website is phished and if genuine, it prompts the user that the website is legitimate.
* We look at the exactness of various classifiers and discovered Random Forest as the best classifiers which gives the most extreme precision.

1. Fig.4. UI of Website Checking Portal
2. However, if the URL entered by a user is found to be a phishing website, a small pop-up will appear on the screen to warn the user regarding this malicious website. There are times when a user needs to access some data on that website, so he/she can select a CONFIRM option to open the website, otherwise he/she will be sent back to the above webpage.
3. Fig.5. Alert Warning for Fraudulent Websites

**RESULTS**

1. Scikit-learn tool has been used to import Machine learning algorithms. Each classifier is trained using training set and testing set is used to evaluate performance of classifiers.
2. Performance of classifiers has been evaluated by calculating classifier's accuracy score.
3. improve the accuracy of our models with better feature extraction.
4. Fig.6. Accuracy with Random Forest Algorithm
5. Fig.7. Accuracy with Decision Tree Algorithm

**CONCLUSION**

1. Thus to summarize, we have seen how phishing is a huge threat to the security and safety of the web and how phishing detection is an important problem domain. We have reviewed some of the traditional approaches to phishing detection; namely blacklist and heuristic evaluation methods, and their drawbacks. We have tested two machine learning algorithms on the Phishing Websites Dataset and reviewed their results. We then selected the best algorithm based on its performance and built a Chrome extension for detecting phishing web pages. The extension allows easy deployment of our phishing detection model to end users. We have detected phishing websites using Random Forest algorithm with and accuracy of 97.31%. For future enhancements, we intend to build the phishing detection system as a scalable web service which will incorporate online learning so that new phishing attack patterns can easily be learned and

**FUTURE SCOPE**

1. Although the use of URL lexical features alone has been shown to result in high accuracy (97%), phishers have learned how to make predicting a URL destination difficult by carefully manipulating the URL to evade detection. Therefore, combining these features with others, such as host, is the most effective approach .
2. For future enhancements, we intend to build the phishing detection system as a scalable web service which will incorporate online learning so that new phishing attack patterns can easily be learned and improve the accuracy of our models with better feature extraction.

**REFERENCES**

1. J. Shad and S. Sharma, A Novel Machine Learning Approach to Detect Phishing Websites Jaypee Institute of Information Technology, pp. 425430, 2018.
2. Y. SÃ¶nmez, T. Tuncer, H. GÃ¶kal, and E. Avci, Phishing web sites features classification based on extreme learning machine, 6th Int. Symp. Digit. Forensic Secur. ISDFS 2018 – Proceeding, vol. 2018 Janua, pp. 15, 2018.
3. T. Peng, I. Harris, and Y. Sawa, Detecting Phishing Attacks Using Natural Language Processing and Machine Learning, Proc. – 12th IEEE Int. Conf. Semant. Comput. ICSC 2018, vol. 2018Janua, pp. 300301, 2018.
4. M. Karabatak and T. Mustafa, Performance comparison of classifiers on reduced phishing website dataset, 6th Int. Symp. Digit. Forensic Secur. ISDFS 2018 – Proceeding, vol. 2018Janua, pp. 15, 2018.
5. S. Parekh, D. Parikh, S. Kotak, and P. S. Sankhe, A New Method for Detection of Phishing Websites: URL Detection, in 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), 2018, vol. 0, no. Icicct, pp. 949952.
6. K. Shima et al., Classification of URL bitstreams using bag of bytes, in 2018 21st Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN), 2018, vol. 91, pp. 15.
7. W. Fadheel, M. Abusharkh, and I. Abdel-Qader, On Feature Selection for the Prediction of Phishing Websites, 2017 IEEE 15th Intl Conf Dependable, Auton. Secur. Comput. 15th Intl Conf Pervasive Intell. Comput. 3rd Intl Conf Big Data Intell. Comput. Cyber Sci. Technol. Congr., pp. 871876, 2017.
8. X. Zhang, Y. Zeng, X. Jin, Z. Yan, and G. Geng, Boosting the Phishing Detection Performance by Semantic Analysis, 2017.