Malicious Website Detection

Aniket Sharma
Department of Computer Engg.,
MIT WPU,
Pune, Maharashtra, India
aniketsharma2468@gmail.com

Sakshi Saindane
Department of Computer Engg.
MIT WPU,
Pune, Maharashtra, India
sakshi.saindane15@gmail.com

Taha Patil
Department of Computer Engg.
MIT WPU,
Pune, Maharashtra, India
tahapatil2001@gmail.com

Ayush Shrivastav
Department of Computer Engg.,
MIT WPU,
Pune, Maharashtra, India
ayushnshrivastav@gmail.com

Pranav Nirmal
Department of Computer Engg.
MIT WPU,
Pune, Maharashtra, India
nirmalpranav187@gmail.com

I. ABSTRACT

'Big Data' refers to large data, data from multiple sources, data with multiple attributes, mainly unstructured/ raw data. Datasets like web logs, sensor logs, healthcare records, document archives, etc are large, complex and mainly semi structured so traditional DBMS methods like SQL cannot be used to process such data. Malicious URLs are one of the easiest ways for successful criminal activities such as phishing, spamming and malware/ trojan downloads. Traditional detection of malicious URLs require the content of the page to be parsed to conclude if the page is malicious or not, which becomes slow. So working with the URLs directly to conclude results in 75% reduction of workload while detecting at least 90% of the malicious URLs. With today's technology where it is very accessible to create hundreds of proxies, this method handles the increasing scale very well.

With advances in technology, there is a constant need of sharing resources in the form of social media, blogs, and through links to websites. To make sharing easier, URL shortening services like bit.ly, goog.gl, tinyurl.com, ow.ly are used, but often lead to unforeseen issues. The seemingly benign short URLs conceal malicious content. For example, a user who visits a malicious website can become a victim of malicious activities such as phishing, spamming, social engineering, and drive-by-download. This project aims to detect malicious shortened URLs with the help of machine learning techniques.

II. INTRODUCTION

'Big Data' refers to large data, data from multiple sources, data with multiple attributes, mainly unstructured/ raw data. Everyday a big amount of data is generated, fueling even further. The rate of data growth is very high and with increase in data there is betterment in the respective services. Uniform Resource Locator (URL) is an identifier used to locate different web addresses. URL is made of 3 major components: a scheme, a host and a path.

URL has a protocol identifier indicating what protocol to use and resource name, used to specify IP address. A major part of Cyber Crime includes Malicious URLs like drive-by-download,

spamming and phishing. Via such sites attackers spread malicious programs or steal personal data. These URLs used by attackers redirects the users to malicious web sites, or phishing sites or downloads malware programs in their system. Some popular techniques used by attackers include: Drive-by Download, Phishing and Social Engineering, and spam. Phishing is the most commonly used social engineering and cyber attack. Through such attacks, the phisher targets naïve online users by tricking them into revealing confidential information, with the purpose of using it fraudulently

In order to avoid getting phished, users should have awareness of phishing websites, have a blacklist of phishing websites which requires the knowledge of website being detected as phishing, detect them in their early appearance, using machine learning and deep neural network algorithms.

Of the above three, the machine learning based method is proven to be most effective than the other methods. Even then, online users are still being trapped into revealing sensitive information in phishing websites

III. OBJECTIVE

A phishing website is a common social engineering method that mimics trustful uniform resource locators (and webpages The objective of this project is to train machine learning models and deep neural nets on the dataset created to predict phishing websites Both phishing and benign URLs of websites are gathered to form a dataset and from them required URL and website content based features are extracted The performance level of each model is measures and compared

IV. LITERATURE SURVEY

Research Paper on Cyber Security

(Contemporary Research In India 2021)
Explains ease of work due to cyber security with different types of cyber security i.e. phishing, ransomware, malware, etc. CIA standards with advantages and disadvantages.
Positive aspect:Offers security to the network or system, and we all know that securing anything has a lot of advantages - protection of complex data, hampering illegal access
Limitations: firewalls can be challenging to configure
correctly, defective configured firewalls might prohibit operators

A Study Of Cyber Security Challenges And Its Emerging Trends On Latest Technologies (researchgate.net 2014)
Trends changing cyber security, top network threats, cyber ethics, cloud computing
Positive aspects: It is a vast topic that is becoming more and more important because world is becoming more interconnected

A MACHINE LEARNING BASED WEB SERVICE FOR MALICIOUS URL DETECTION IN A BROWSER
(Faculty of Purdue University, 2019)
Applied different Machine Learning algorithms on Malicious URLs on different web servers and classified them.
Positive aspects: Variety of algorithms applied giving a broad view on malicious URLs. Gives knowledge about attacks on different web servers.

Malicious URL filtering — A big data application
(2013 IEEE conference)
Gives a basic comparison of traditional filtering and URL based filtering
Positive aspects: Provides great level 0 understanding of URL filtering
Limitations: Limited to the URL provided and not its proxies/ redirects

Detection of malicious URLs in big data using RIPPER Algorithm
(IJSER © 2020)
Gives an overview of machine learning/ non machine learning approaches to filter URL,

RIPPER Algorithm
Positive aspect: Proper usage of RIPPER algorithm and blacklisting technique used
Limitations: Training set is static and is limited to 600 URLS (400 malicious, 200 Legitimate)

Malicious URL Detection based on Machine Learning
(IJACSA 2020)
Ease to calculate attributes and big data processing technologies together used to ensure balance of the two factors in processing time and accuracy.
Using RNN- LSTM, random forests the URL data is classified.
Positive aspect: Usage of real time data and using neural Networks for better accuracy.

A hybrid DNN–LSTM model for detecting phishing URLs
(Springer-Verlag London 2021)
Emphasis on detecting phishing URLs using DNN-LSTM (NLP). In depth analysis with comparison of algorithms, including NLP features

V. APPROACH

Completion of this project:

- Collect dataset containing phishing and legitimate websites from the open source platforms.
- Write a code to extract the required features from the URL database.
- Analyse and preprocess the dataset by using EDA techniques.
- Divide the dataset into training and testing sets.
- Run selected machine learning and deep neural network algorithms like SVM, Random Forest, Autoencoder on the dataset.
- Write a code for displaying the evaluation result considering accuracy metrics.

formats like csv, json etc that gets updated hourly

Form the obtained collection, 5000 URLs are randomly picked

Positive aspect: Detecting malicious URLs using lexical, host-based and content-based features

Classifying Phishing URLs Using Recurrent Neural Networks
(MindLab Research Group, Universidad Nacional de Colombia, Bogota´ 2017)

Malicious URL Detection based on Machine Learning.
(IJACSA 2020)
Positive aspect: Ease to calculate attributes and big data processing technologies together used to ensure balance of the two factors in processing time and accuracy.
Detecting malicious URLs using lexical, host-based and content-based features

Below mentioned are the steps involved in the

- Compare the obtained results for trained models and specify which is better.

VI. DATA COLLECTION

Legitimate URLs are collected from the dataset provided by University of New Brunswick, https://www.unb.ca/cic/datasets/url_2016.html

From the collection, 5000 URLs are randomly picked

Phishing URLs are collected from open source service called PhishTank This service provide a set of phishing URLs in multiple

VII. FEATURE SELECTION

The following category of features are selected:

- Address Bar based Features

  The address bar is the familiar text field at the top of a web browser's graphical user interface (GUI) that displays the name or the URL (uniform resource locator) of the current web page. Users request websites and pages by typing either the name or the URL into the address bar.

- Domain based Features

  Domain based features contains ".com" effect, keeping it short, easy to remember, perfect match with site niche, Easy to spell, using keywords, avoiding awkward spelling mistakes.

Address Bar based Features considered are:

- Domain of URL

  Here, we are just extracting the domain present in the URL. This feature doesn't have much significance in the training. May even be dropped while training the model.

- Redirection '//' in URL

  Checks the presence of "//" in the URL. The existence of "//" within the URL path means that the user will be redirected to another website. The location of the "//" in URL is computed. We find that if the URL starts with "HTTP", that means the "//" should appear in the sixth position. However, if the URL employs "HTTPS" then the "//" should appear in seventh position.

  If the "//" is anywhere in the URL apart from after the protocal, thee value assigned to this feature is 1 (phishing) or else 0 (legitimate).

- IP Address in URL

  Checks for the presence of IP address in the URL. URLs may have IP address instead of domain name. If an IP address is used as an alternative of the domain name in the URL, we can be sure that someone is trying to steal personal information with this URL.

  If the domain part of URL has IP address, the value assigned to this feature is 1 (phishing) or else 0 (legitimate).

- ' https' in Domain name

  Checks for the presence of "http/https" in the domain part of the URL. The phishers may add the "HTTPS" token to the domain part of a URL in order to trick users.

  If the URL has "http/https" in the domain part, the value assigned to this feature is 1 (phishing) or else 0 (legitimate).

- '@' Symbol in URL

  Checks for the presence of '@' symbol in the URL. Using "@" symbol in the URL leads the browser to ignore everything preceding the "@" symbol and the real address often follows the "@" symbol.

  If the URL has '@' symbol, the value assigned to this feature is 1 (phishing) or else 0 (legitimate).

- Using URL Shortening Service

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. This is accomplished by means of an "HTTP Redirect" on a domain name that is short, which links to the webpage that has a long URL.

If the URL is using Shortening Services, the value assigned to this feature is 1 (phishing) or else 0 (legitimate).

- Length of URL

  Computes the length of the URL. Phishers can use long URL to hide the doubtful part in the address bar. In this project, if the length of the URL is greater than or equal 54 characters then the URL classified as phishing otherwise legitimate.

  If the length of URL >= 54 , the value assigned to this feature is 1 (phishing) or else 0 (legitimate).

- Depth of URL

  Computes the depth of the URL. This feature calculates the number of sub pages in the given url based on the '/'.

  The value of feature is a numerical based on the URL.

- Prefix or Suffix " "--" in Domain

  Checking the presence of '-' in the domain part of URL. 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.

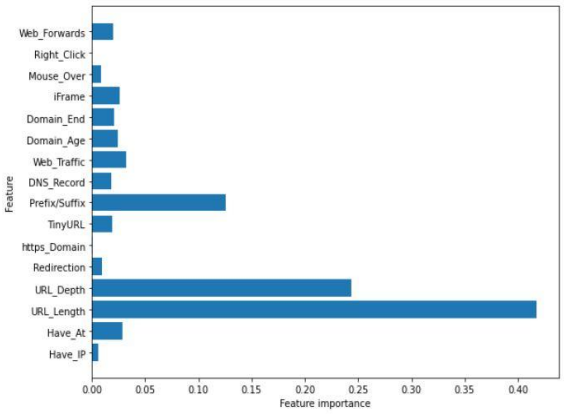
  If the URL has '-' symbol in the domain part of the URL, the value assigned to this feature is 1 (phishing) or else 0 (legitimate).

Domain based Features considered are:

- DNS Record

  DNS records (aka zone files) are instructions that live in authoritative DNS servers and provide information about a domain including what IP address is associated with that domain and how to handle requests for that domain.

  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 value assigned to this feature is 1 (phishing) or else 0 (legitimate).



VII.I. Visualising the data

- Age of Domain

  This feature can be extracted from WHOIS database. Most phishing websites live for a short period of time. The

minimum age of the legitimate domain is considered to be 12 months for this project. Age here is nothing but different between creation and expiration time.

If age of domain > 12 months, the vlaue of this feature is 1 (phishing) else 0 (legitimate).

- Website Traffic

  This feature measures the popularity of the website by determining the number of visitors and the number of pages they visit. However, since phishing websites live for a short period of time, they may not be recognized by the Alexa database (Alexa the Web Information Company., 1996). By reviewing our dataset, we find that in worst scenarios, legitimate websites ranked among the top 100,000. Furthermore, if the domain has no traffic or is not recognized by the Alexa database, it is classified as “Phishing”.

  If the rank of the domain < 100000, the vlaue of this feature is 1 (phishing) else 0 (legitimate).

- End Period of Domain

  This feature can be extracted from WHOIS database. For this feature, the remaining domain time is calculated by finding the different between expiration time & current time. The end period considered for the legitimate domain is 6 months or less for this project.

  If end period of domain > 6 months, the vlaue of this feature is 1 (phishing) else 0 (legitimate).

HTML and JavaScript based Features considered are:

- Iframe Redirection

  IFrame is an HTML tag used to display an additional webpage into one that is currently shown. Phishers can make use of the “iframe” tag and make it invisible i.e. without frame borders. In this regard, phishers make use of the “frameBorder” attribute which causes the browser to render a visual delineation.

  If the iframe is empty or repsonse is not found then, the value assigned to this feature is 1 (phishing) or else 0 (legitimate).

- Disabling Right Click

  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”. Nonetheless, for this feature, we will search for event “event.button==2” in the webpage source code and check if the right click is disabled.

  If the response is empty or onmouseover is not found then, the value assigned to this feature is 1 (phishing) or else 0 (legitimate).

- Status Bar Customization

  Phishers may use JavaScript to show a fake URL in the status bar to users. To extract this feature, we must dig-out the webpage source code, particularly the

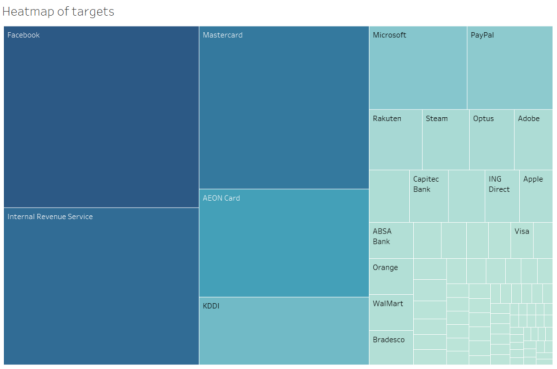
"onMouseOver" event, and check if it makes any changes on the status bar

If the response is empty or onmouseover is found then, the value assigned to this feature is 1 (phishing) or else 0 (legitimate).

- Website Forwarding

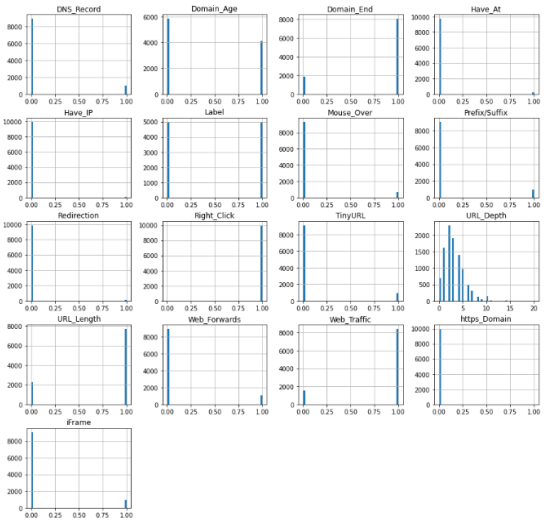
  The fine line that distinguishes phishing websites from legitimate ones is how many times a website has been redirected. In our dataset, we find that legitimate websites have been redirected one time max. On the other hand, phishing websites containing this feature have been redirected at least 4 times.

All together 17 features are extracted from the dataset.



VII.II. Heatmap of Targets

VIII. FEATURE DISTRIBUTION



IX. MODEL AND TRAINING

This is a supervised machine learning task. There are two major types of supervised machine learning problems, called classification and regression.

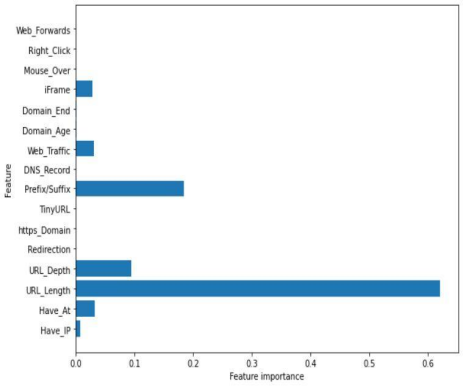
This data set comes under classification problem as the input URL is classified as phishing (1) or legitimate (0). The machine learning models (classification) considered
to train the dataset in this notebook are:

- Decision Tree

  Decision trees are widely used models for classification and regression tasks. Essentially, they learn a hierarchy of if/else questions, leading to a decision. Learning a decision tree means learning the sequence of if/else questions that gets us to the true answer most quickly.

  In the machine learning setting, these questions are called tests (not to be confused with the test set, which is the data we use to test to see how generalizable our model is). To build a

tree, the algorithm searches over all possible tests and finds the one that is most informative about the target variable.

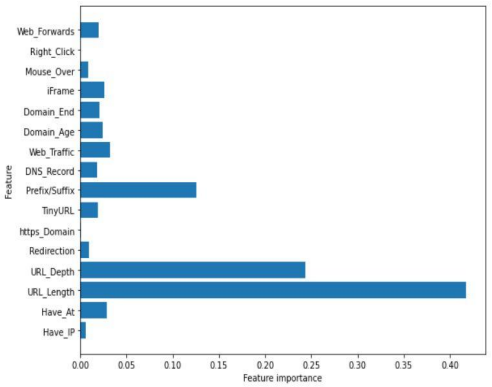


IX.I. Feature importance in Decision Tree Model

- Random Forest

Random forests for regression and classification are currently among the most widely used machine learning methods.A random forest is essentially a collection of decision trees, where each tree is slightly different from the others. The idea behind random forests is that each tree might do a relatively good job of predicting, but will likely overfit on part of the data.

If we build many trees, all of which work well and overfit in different ways, we can reduce the amount of overfitting by averaging their results. To build a random forest model, you need to decide on the number of trees to build (the n_estimators parameter of RandomForestRegressor or RandomForestClassifier). They are very powerful, often work well without heavy tuning of the parameters, and don't require scaling of the data.



IX.II. Feature importance in Random Forest Model

- Multilayer Perceptrons

Multilayer perceptrons (MLPs) are also known as (vanilla) feed-forward neural networks, or sometimes just neural networks. Multilayer perceptrons can be applied for both classification and regression problems.

MLPs can be viewed as generalizations of linear models that perform multiple stages of processing to come to a decision.

- XGBoost

XGBoost is one of the most popular machine learning algorithms these days. XGBoost stands for eXtreme Gradient Boosting. Regardless of the type of prediction task at hand; regression or classification. XGBoost is an implementation of gradient boosted decision trees designed for speed and performance.

- Autoencoder Neural Network

An auto encoder is a neural network that has the same number of input neurons as it does outputs. The hidden

layers of the neural network will have fewer neurons than the input/output neurons. Because there are fewer neurons, the auto-encoder must learn to encode the input to the fewer hidden neurons. The predictors (x) and output (y) are exactly the same in an auto encoder.

- Support Vector Machines

  In machine learning, support-vector machines (SVMs, also support-vector networks) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier.

X. MODEL EVALUATION

The models are evaluated, and the considered metric is accuracy.

Below Figure shows the training and test dataset accuracy by the respective models:

	ML Model	Train Accuracy	Test Accuracy
3	XGBoost	0.868	0.857
2	Multilayer Perceptrons	0.866	0.854
4	AutoEncoder	0.810	0.810
1	Random Forest	0.820	0.809
0	Decision Tree	0.816	0.803
5	SVM	0.806	0.786

X.1. Analysing model performance

For the above it is clear that the XGBoost model gives better performance. The model is saved for further usage.

XI. FUTURE SCOPE

Working on this project is very knowledgeable and worth the effort. Through this project, one can know a lot about the phishing websites and how they are differentiated from legitimate ones.

This project can be taken further by creating a browser extensions of developing a GUI.

These should classify the inputted URL to legitimate or phishing with the use of the saved model.

XII. CONCLUSION

Malicious websites have become very prominent these days and threatened the security of the Internet. Machine learning approach is most effective for the detection of malicious URLs when compared to general blacklisting technique. Detecting malicious websites before we get affected in one or other way is very crucial. Keeping that in the mind, with the availability of many URL shortening services, there are no prior methods that classify the short URLs as malicious or benign.

Before stating the ML model training, the data is split into 80-20 i.e., 8000 training samples & 2000 testing samples. From the dataset, it is clear that this is a supervised machine learning task. There are two major types of supervised machine learning problems, called classification and regression.

This data set comes under classification problem, as the input URL is classified as phishing (1) or legitimate (0). The supervised machine learning models (classification) as considered to train the dataset in this project are:

- Decision Tree
- Random Forest
- Multilayer Perceptrons
- XGBoost
- Autoencoder Neural Network
- Support Vector Machines

All these models are trained on the dataset and evaluation of the model is done with the test dataset.

XIII. REFERENCE

[1] Detection of Malicious URLs in Big Data Using Ripper Algorithm "https://www.ijser.org/researchpaper/DETECTION-OF-MALICIOUS-URLs-IN-BIG-DATA-USING-RIPPER-ALGORITHM.pdf"

[2] Malicious URL Detection based on Machine Learning "https://thesai.org/Downloads/Volume11No1/Paper_19-Malicious_URL_Detection_based_on_Machine_Learning.pdf"

[3] Malicious URL Filtering – A Big Data Application "https://jupiter.math.nycu.edu.tw/~yuhjye/assets/file/publications/conference_papers/C6_Malicious%20URL%20Filtering%20-%20A%20Big%20Data%20Application.pdf"

[4]Improving malicious URLs detection via feature engineering: Linear and nonlinear space transformation methods "https://www.sciencedirect.com/science/article/pii/S0306437920300053"

[5] Detection of malicious URLs in big data using RIPPER algorithm "https://ieeexplore.ieee.org/document/8256808"

[6]M. Darling, "A Lexical Approach for Classifying Malicious URLs", 2015. [Online]. Available: http://digitalrepository.unm.edu/ece_etds/63. [Accessed: Oct 2017].