APPLIED DATA SCIENCE INTERNSHIP

PROJECT REPORT

PROJECT TITLE:

“WEB PHISHING DETECTION USING IBM WATSON STUDIO”

TEAM MEMBERS:

G VISHNU SAI (18BLC1079)

Y SAI PAVAN (18BLC1148)

CH VINEETH (18BEC1327)

M SAI MANIKANTA (18BEC1314)

INDEX

1. Introduction 3
   a. Overview
   b. Purpose
2. Literature Survey 4
   a. Existing problem
   b. Proposed solution
3. Theoretical analysis 6
   a. Block diagram
   b. Software designing
4. Experimental investigations 8
5. Flowchart 9
6. Result 10
7. Advantages 11
8. Applications 12
9. Conclusion 12
10. Future Scope 13
11. Bibliography 13
12. Appendix 14
   a. Source code
   b. UI output screenshot

1. INTRODUCTION

a. Overview

The goal of our project is to implement a machine learning solution to the problem of detecting phishing and malicious web links. The end result of our project will be a software product which uses machine learning algorithm to detect malicious URLs. Phishing is the technique of extracting user credentials and sensitive data from users by masquerading as a genuine website. In phishing, the user is provided with a mirror website which is identical to the legitimate one but with malicious code to extract and send user credentials to phishers. Phishing attacks can lead to huge financial losses for customers of banking and financial services. The traditional approach to phishing detection has been to either to use a blacklist of known phishing links or heuristically evaluate the attributes in a suspected phishing page to detect the presence of malicious codes. The heuristic function relies on trial and error to define the threshold which is used to classify malicious links from benign ones. The drawback to this approach is poor accuracy and low adaptability to new phishing links. We plan to use machine learning to overcome these drawbacks by implementing some classification algorithms and comparing the performance of these algorithms on our dataset.

b. Purpose:

The main purpose of the project is to detect the fake or phishing websites who are trying to get access to the sensitive data or by creating the fake websites and trying to get access of the user personal credentials. We are using machine learning algorithms to safeguard the
sensitive data and to detect the phishing websites who are trying to gain access on sensitive data.

2. LITERATURE SURVEY

a. Existing Problem

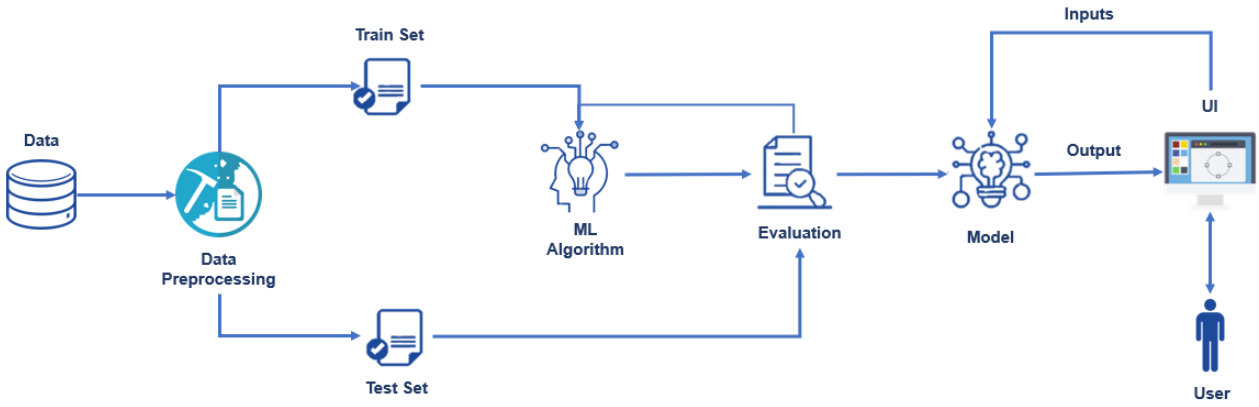
Phishing is the most commonly used social engineering and cyber-attack. Through such attacks, the phisher targets naïve online users by tricking them into revealing confidential information, with the purpose of using it fraudulently. In order to avoid getting phished, users should have awareness of phishing 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.

b. Proposed Solution

This section describes the proposed model of phishing attack detection. The proposed model focuses on identifying the phishing attack based on checking phishing websites features, Blacklist and WHOIS database. According to few selected features can be used to differentiate between legitimate and spoofed web pages. These selected features are many such as URLs, domain identity, security & encryption, source code, page style and contents, web address bar and social human factor. This study focuses only on URLs and domain name features. Features of URLs and domain names are checked using several criteria such as IP Address, long URL address, adding a prefix or suffix, redirecting using the symbol "//", and URLs having the symbol "@".These features are inspected using a set of rules in order to distinguish URLs of phishing webpages from the URLs of legitimate websites

3. THEORETICAL ANALYSIS

a. Block Diagram



b. Software Designing

- Selection of programming language : **Python**

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together.

i) Jupyter Notebook :

The Jupyter Notebook App is a server-customer application that permits altering and running note pad records by means of an internet browser. The Jupyter Notebook App can be executed on a nearby work area requiring no web access (as portrayed in this report) or can be introduced on a remote server and got to through the web. Notwithstanding showing/altering/running note pad archives, the

Jupyter Notebook App has a "Dashboard" (Notebook Dashboard), a "control board" indicating nearby records and permitting to open note pad reports or closing down their portions.

ii) Sklearn:

Scikit-learn is a library in Python that provides many unsupervised and supervised learning algorithms.

iii) NumPy:

NumPy is a Python package that stands for 'Numerical Python'. It is the core library for scientific computing, which contains a powerful n-dimensional array object.

iv) Pandas:

pandas is a fast, powerful, flexible, and easy to use open source data analysis and manipulation tool, built on top of the Python programming language.

v) Matplotlib:

It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits

vi) Flask:

Flask is an API of Python that allows us to build up web-applications.

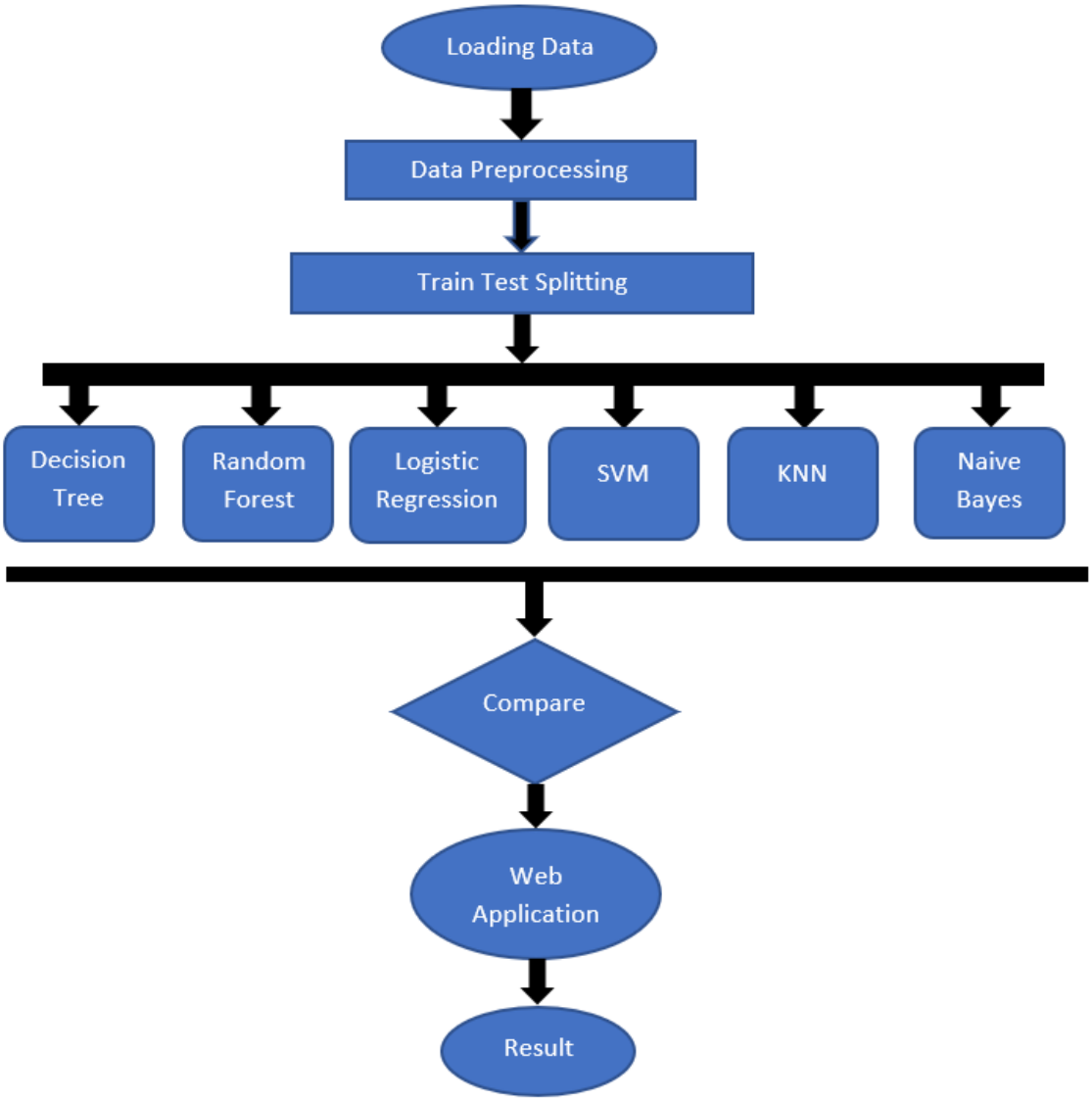
4. EXPERIMENTAL INVESTIGATIONS

From the data given it is a supervised machine learning task. In that dataset comes under classification problem, as the input

URL is classified as phishing (1) or legitimate (0). The machine learning models considered 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.
- 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.
- KNN:
  K Nearest Neighbor algorithm falls under the Supervised Learning category and is used for classification (most commonly) and regression. It is a versatile algorithm also used for imputing missing values and resampling datasets.
- Naïve Bayes:
  Naive Bayes is a probabilistic machine learning algorithm based on the Bayes Theorem, used in a wide variety of classification tasks. In this post, you will gain a clear and complete understanding of the Naive Bayes algorithm and all necessary concepts so that there is no room for doubts or gap in understanding.
- SVM:
  A Support Vector Machine (SVM) is a supervised machine learning algorithm that can be employed for both classification and regression purposes. SVMs are more commonly used in classification problems.
- Logistic Regression:
  Logistic regression is a supervised learning classification algorithm used to predict the probability of a target variable.

5. FLOWCHART


Loading Data
Data Preprocessing
Train Test Splitting
Decision Tree
Random Forest
Logistic Regression
SVM
KNN
Naive Bayes
Compare
Web Application
Result

6. RESULT

For the given data we analyzed and preprocessed dataset by using EDA techniques and removed outliners using Zscore

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

'K-nearest neighbors' is the best fit model for the given dataset with **'96.3'** accuracy

we removed outliners using both methods Zscore, IQR and applied to models and compared the accuracy's we got better results for Zscore removed outliners

we have used GridSerachCv for tuning the parameters and applied best params to the models which gave us better results

7. ADVANTAGES & DISADVANTAGES

a. Advantages :

- Email filtering solutions help in filtering phishing/spam emails, but this provides holistic protection for all outgoing internet traffic.
- Centralized solution implemented org-wide and no dependency on client-side agents/software.
- Data mining algorithm used in this system provides better performance as compared to other traditional classifications algorithms.
- This system can be used by many E-commerce or other websites in order to have good customer relationship.
- Reduce dependency, cost & license on third-party external software.
- Detect and prevent against unknown phishing attacks, as new patterns are created by attackers.
- Real-time protection for employees who access malicious websites or click on phishing links.
- Next level of intelligence on top of signature-based prevention techniques & blacklists

b. Disadvantages :

- Process is not accuracy.
- All websites related data will be stored in one place.
- It will assessment slowly.
- If Internet connection fails, this system won't work.
- It will take time to load all the dataset.
- All e-banking websites related data will be stored in one place.

8. APPLICATIONS

- This framework will be helpful for some Web based business undertakings.
- This framework will be helpful for some clients who buy items on the web.
- Cyber Security Analysts can use the feature extraction component to quickly analyze indicators and hence expedite incident response.
- Helps security designers to assemble more intelligent items, customized to their own organization necessities.
- Provide insights into building an ML pipeline, data engineering & feature extraction
- Figure out how to fingerprint a URL for phishing indicators using various data sources and components
- Figure out how to retrain the model for better exactness and importance

9. CONCLUSION

Through this project, one can know a lot about the phishing websites and how they are differentiated from legitimate ones. We have trained machine learning models on the dataset created to predict phishing websites, this gives us good understanding to apply machine learning on real time applications. We have learned to integrate a web application using the flask framework and learned basics of web scripting languages. We have learned to train models on IBM Watson studio.

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. We have tested three machine learning algorithms on

the ‘Phishing Websites Dataset’ from the UCI Machine Learning Repository 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.

10. FUTURE SCOPE

Although the use of URL lexical features alone has been shown to result in high accuracy (96.3%), 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 .

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.

11. BIBLIOGRAPHY

i) https://github.com/citizenlab/test-lists

ii) https://ieeexplore.ieee.org/document/8474820

iii) https://ieeexplore.ieee.org/document/9218076/

iv) https://ieeexplore.ieee.org/abstract/document/8004877/

v) https://www.ijert.org/detection-of-phishing-websites-using-machine-learning

vi) Anti-Phishing Working Group. Phishing Activity Trends.Report,http://antiphishing.org/apwg_report_final.pdf. 2007.

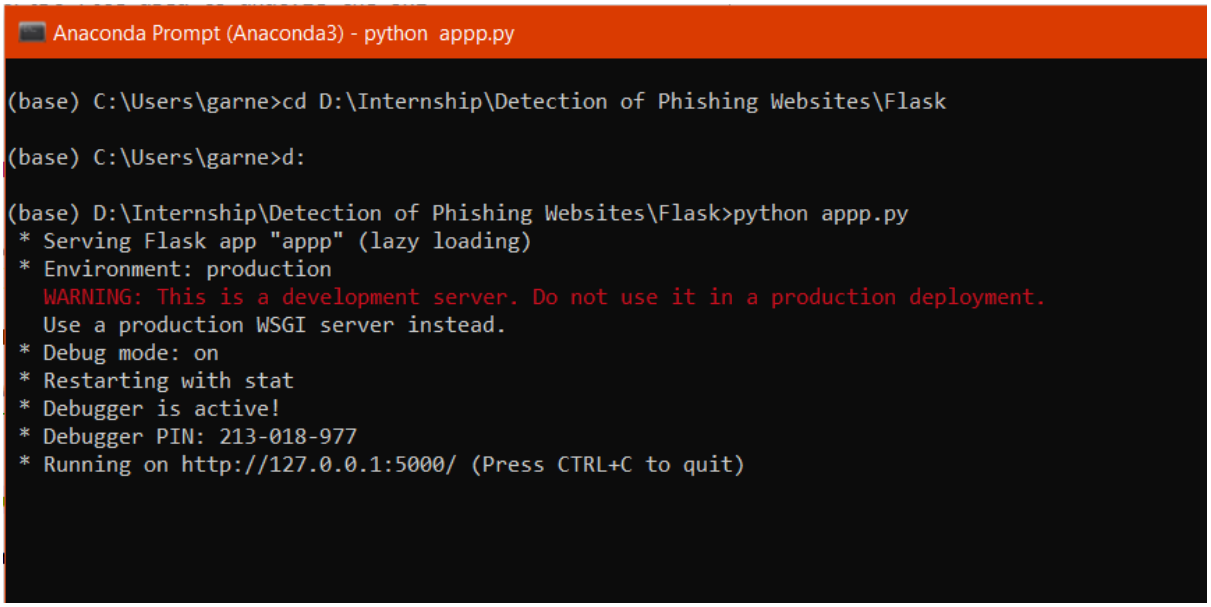
vii) FDIC., “Putting an End to Account-Hijacking Theft,”

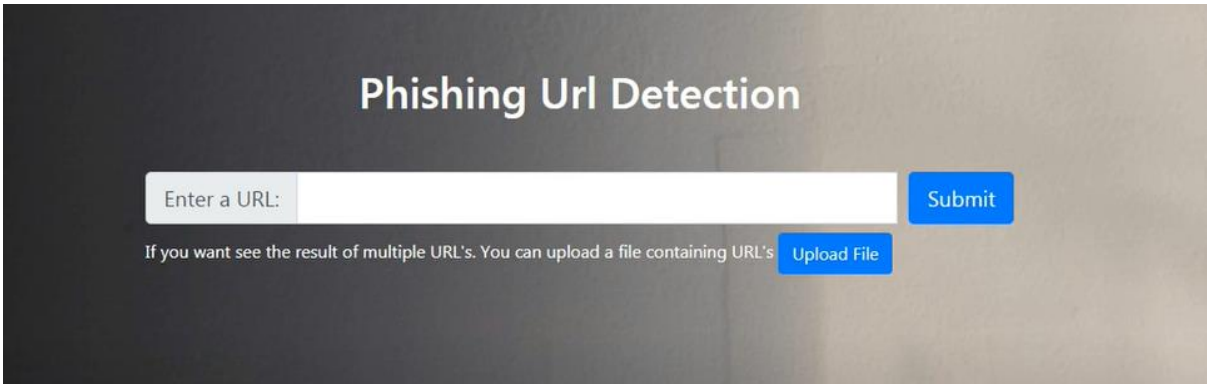
viii) http://wwfdic.gov/consumers/id/identity_theft.pdf, 2004.

12. APPENDIX

a. Source Code – GitHub

b. UI Output screenshot:





C. Results Screenshots

Models' accuracy

	ML Model	Train Accuracy	Test Accuracy
2	KNN	0.987	0.963
1	Random Forest	0.949	0.952
0	Decision Tree	0.954	0.938
4	SVM	0.924	0.921
5	Logistic Regression	0.924	0.921
3	Naive Bayes	0.623	0.612

Visualizing models accuracy