International Journal For Advanced Research In Science & Technology
A peer reviewed international journal www.ijarst.in
ISSN: 2457-0362

Predicting phishing websites based on Build classification models of Supervised Machine Algorithms

[1]Gottapu Dhanunjaya Rao & [2]Narlakanti Harsha Vardhan
[1,2]UG, Department of Computer Science Engineering
SR International Institute of Technology, Hyderabad, Telangana 501301

Abstract

Phishing are one of the vital customary and most dangerous attacks amongst cybercrimes. The aim of those assaults is to steal the knowledge used by members and businesses to behavior transactions. Phishing web sites include quite a lot of recommendations among their contents and net browser-based know-how. The intent of this learn is to participate in severe finding out machine (ELM) 75 % headquartered classification for 30 facets together with Phishing internet sites knowledge in UC Irvine computing device studying Repository database. For outcome evaluation, ELM was once compared with other desktop studying methods such as support Vector computing device (SVM) 76 %, Naïve Bayes (NB) 71 % and detected to have the perfect accuracy of 76 %.

Keywords: - url action, Prediction, SVM, Naïve Bayes, ELM

1. INTRODUCTION

In the internet world, visiting different web pages becomes our daily activity for various reasons. In the internet world, so many legitimate websites provide contributions to knowledge sharing, research, entertainment, etc. Another side, a few websites cause security issues to internet users. These websites imitate legitimate websites and steal personal and professional information from the users. Identification of phishing websites is an important research topic in the Cyber Security research domain. In this research, many different types of systems were proposed for the detection of phishing websites. Sahingoz et al. (2019) proposed a Phishing Websites Detection system using NLP features, namely, Raw Word Count, Domain Verification, Average Word Count, etc. In this work, they used both text data and a few context data of the websites and implemented Machine Learning algorithms for classifications. This proposed system depends on the features of the websites for predicting phishing websites. Zouina et al. (2017) implemented detection of phishing websites using features of the website. This work used supervised Machine Learning algorithms of Support Vector Machine for prediction, but this work using only 6 features of the websites. Similarly, Ferreira et al. (2018) proposed a Feature-based phishing prediction model with a Neural Network algorithm. Identification of phishing websites is a popular research topic in the Cyber Security domain. Many studies have been conducted to identify phishing sites using various types of data, such as web text, web context, web features, and so on. Identification of phishing websites using text data is more popular. Using data mining or machine learning algorithms, the text data is trained and predicted. Recently, Adewole et al. (2021) proposed a system of prediction for the identification of phishing websites using web text data. For these reasons, this work focused on features-based and text-based approaches for detecting phishing websites.

2. METHODOLOGY

This work proposed a system to predict phishing websites, based on the two models, namely, Feature-based, and Text-based. This section explained the methodology of the proposed system with an architecture diagram, illustrated in Figure 3.1. The architecture of the proposed system can diverge into variants, namely, Classification analysis and user-side prediction. The classification analysis depends on two types of data variants, those are, Feature-based and Text-based. This section described the main modules of the methodology.

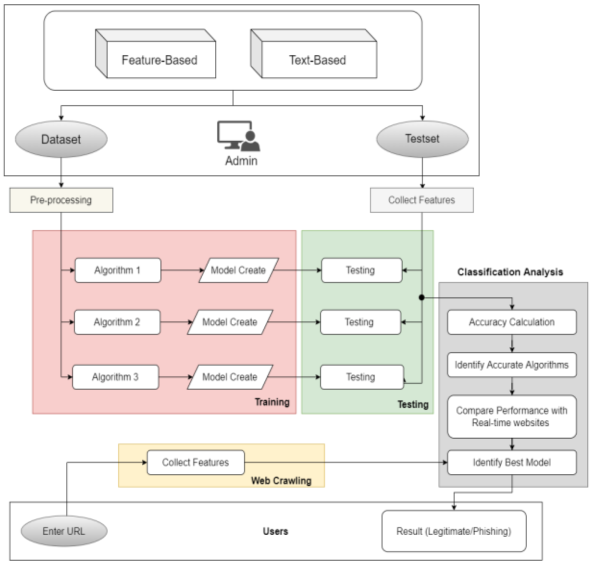


Figure 1: Proposed Architecture

Architecture Overview

The main goals of this work are, to identify the best model among the text-based and feature-based models and identify the most accurate classification model for the prediction of phishing websites. To achieve these goals, need to collect two types of datasets of feature-based and test-based, and implement the classification analysis for both datasets to determine the adequate ML model. After determining the models of both models, identify the most accurate model with a few malware and legitimate websites which were collected manually. The architecture is designed with two stakeholders, namely, the admin and users. The admin is responsible for determining the suitable and accurate ML model for the prediction of phishing websites based on the classification analysis, The users can see the prediction results by providing the inputs of web URLs. The flow of the architecture and main modules of the methodology is discussed in the following sections.

Feature-based model

For prediction of something (e.g., weather prediction, disease prediction, etc) needs to analyze the input data, the input data may have in multiple formats such as text-based, image-based, feature-based, etc. This proposed work evaluated feature-based and text-based data formats. The feature-based model considered features or context data of the web pages. The features of the websites such as URL length, count of the symbols like '.', '//', etc., HTTP or HTTPS token, etc. To meet the requirements of the Feature-based classification analysis of the websites, have taken a feature-based dataset (Rami, 2015) which consists of the Features of the websites labeled with phishing class or legitimate class.

Dataset

The feature-based model used the "Phishing Websites Data Set" (Rami, 2015) for the classification and prediction of phishing websites. This dataset is published in the UCI repository [Link]. This dataset is available in the CSV file format and labeled with two classes, phishing (1) and legitimate (0).

Description of the dataset

This section described the "Phishing Websites Data Set" using Pandas Python API, illustrated in Table 3.1. This dataset has 18 columns with 17 features of websites and is labeled with class data (Phishing website or legitimate website). Each column has 11055 rows data of integer data (0 and 1).

Classification Models

In the classification analysis, this work implemented three Machine Learning models. The classification analysis is the process of enforcing Training and testing for computing the performance results. The Phishing Website Features dataset is separated into two parts in the ratio of 70:30. 70% of the Phishing Website Features dataset is taken for training and 30% of the dataset is used for testing. Based on the testing results, this work calculates the performance results. The classification models used for this dataset are mentioned in the following.

- Naïve Bayes
- Support Vector Machine
- Neural Networks

Implementation of Naïve Bayes Algorithm

The Naive Bayes algorithm is a supervised machine learning algorithm, this algorithm works based on the Bayes theorem [Rohit Dwivedi, (2020)]. To implement this classification model, used Scikit-Learn Python Library. Multiple Naïve Bayes models are available in Scikit-Learn Python Library, for this classification used Bernoulli Naïve Bayes algorithm for training the features data. The implementation of the Naïve Bayes algorithm is mentioned in. In preprocessing, the dataset is loaded with help of Pandas API and converted into Array format with Numpy. The separated features data ‘X’ and class data ‘Y’ are trained with the fit() function.

Implementation of Support Vector Machine Algorithm

The Support Vector Machine (SVM) algorithm is a popular machine learning algorithm, this algorithm discovers the decision boundaries using the hyper line concept [Sunil Ray, (2017)]. To implement this classification model, used Scikit-Learn Python Library. Multiple SVM models are available in Scikit-Learn Python Library, for this classification used Linear SVC algorithm for training the features data. The implementation of the SVM algorithm is mentioned in Figure 3.3. In preprocessing, the dataset is loaded with help of Pandas API and converted into Array format with Numpy. The separated features data ‘X’ and class data ‘Y’ are trained with the fit() function.

Implementation of Neural Network Algorithm

The Neural Network algorithm is also called Artificial Neural Network (ANN), this algorithm works based on the nonlinear activation function in the hidden layers [Pulkit Sharma, (2018)]. To implement this classification model, used Scikit-Learn Python Library. For this classification used a Multi-layer Perceptron classifier algorithm for training the feature data. The implementation of the ANN algorithm is mentioned in Figure 3.4. In preprocessing, the dataset is loaded with help of Pandas API and converted into Array format with Numpy. The separated features data ‘X’ and class data ‘Y’ are trained with the fit() function.

Text-based model

This proposed work considered both feature-based and text-based data formats. The text-based model considered content or webpage text of the web pages. To analyze the text of the phishing and legitimate websites, have taken a text-based dataset (Alex, 2020) which consists of text of the websites and labeled with phishing class or legitimate class.

Dataset

The text-based model used the dataset of the web content (Alex, 2020) for the classification and prediction of phishing websites and legitimate websites. This dataset is published in the Kaggle repository [Link]. This dataset is available in the CSV file format and labeled with two classes, phishing (‘bad’) and legitimate (‘bad’).

Description of the dataset

This section described the dataset of the web content using Pandas Python API, illustrated in Table 3.2. This dataset has 2 columns and is labeled with class data (good website or bad website). Each column has 361934 rows data of String data (Web Text and Class).

Classification Models

In the classification analysis of the text-based model, also implemented three Machine Learning models. The dataset of web content dataset is separated into two parts in the ratio of 70:30. 70% of the web content dataset is taken for training and 30% of the dataset is used for testing. Based on the testing results, this work calculates the performance results. The classification models used for this dataset are mentioned in the following.

- Naïve Bayes
- Support Vector Machine
- Neural Networks

Pre-Processing steps

The Machine Learning classification models depend on Mathematical Approaches. It allows numerical input data for training the data. For text data classification, need to convert the text form to numerical form in pre-process stage. In this conversion, the text content is going to form as words, tokens, or n-gram tokens. In this stage can also implement StopWords removal to reduce the size of the text. After implementing tokenization, should convert the text to numerical form. There are two methods to convert the text to numeric, namely, based on the count of the words and based on the weightage of the words in the statement, documents, etc. Compare to the count of the words, the weightage of the words is the most efficient way to implement the text conversion. In this work implemented 'TfidfVectorizer' for conversion of the text to numerical data for classification. The implementation of 'TfidfVectorizer' using Python Coding is represented sample code of TfidfVectorizer mentioned in, used stop words removal, and uni-gram (n=1). [Unnikrishnan, (2021)]

Implementation of Naïve Bayes Algorithm

The Naive Bayes algorithm is a supervised machine learning algorithm, this algorithm works based on the Bayes theorem [Rohit Dwivedi, (2020)]. To develop this classification model, used Scikit-Learn Python Library. Different types of Naïve Bayes models are available present in the Scikit-Learn Python Library. For this text data classification, used the Multinomial Naïve Bayes algorithm. To convert text data to the numerical matrix, used TfidfVectorizer in the pre-processing stage. The implementation of the Naïve Bayes algorithm is mentioned in Figure 3.6. The text content and class label data are trained with the fit() function and saved as training object in a physical file for reusability.

Implementation of Support Vector Machine Algorithm

The SVM algorithm is a popular supervised machine learning algorithm, this algorithm works based on decision boundaries using the hyper line concept [Sunil Ray, (2017)]. To develop this classification model, used Scikit-Learn Python Library. Different types of SVM models are available present in the Scikit-Learn Python Library. For this text data classification, used LinearSVC (Support Vector Classifier) algorithm. To convert text data to the numerical matrix, used TfidfVectorizer in the pre-processing stage. The implementation of the SVM algorithm is mentioned in Figure 3.7. The text content and class label data are trained with the fit() function and saved as training object in a physical file for reusability.

Implementation of Neural Network Algorithm

The Neural Network algorithm is a supervised machine learning algorithm and this algorithm works based on the nonlinear activation function in the hidden layers [Pulkit Sharma, (2018)]. To develop this classification model, used Scikit-Learn Python Library. For this text data classification, used Multi-layer Perceptron classifier (MLPClassifier). To convert text data to the numerical matrix, used TfidfVectorizer in the pre-processing stage. The implementation of the Neural Network algorithm is mentioned in Figure 3.8. The text content and class label data are trained with the fit() function and saved as training object in a physical file for reusability.

Performance Measures for Feature-based and Text-based classification models

It is important to calculate the performance measures after completion training and testing phase. The calculation of the performance measuring process is conducted for both Text-based and Feature-based classification models. After the testing process, collect the testing data inputs (Actual data) and predicted results. This performance measure used Accuracy calculation for comparing the outcomes of the classification models.

Prediction module

This module is implements at the user portal. After analyzing the performance results between feature-based and text-based results, the system is built with a prediction module with the best classifier. The collects the input URL from the user and based on the prediction model system crawls the input data from the web. Based on the input data, the prediction module predicts the result and presents it to the user.

3. RESULTS

Feature-based Model Page

The Feature-based model page can redirect from the main menu of the admin portal, after verifying the admin session, the system allows to access this page. The feature-based model considered features or context data like length, symbols, etc. of the web pages. This page has three inputs to initiate the training and testing process for the three classifiers. The classification results can see on this page by clicking on the 'View Results' button. As illustrated in, the feature-based model page has multiple inputs to perform classification analysis on a feature-based dataset.

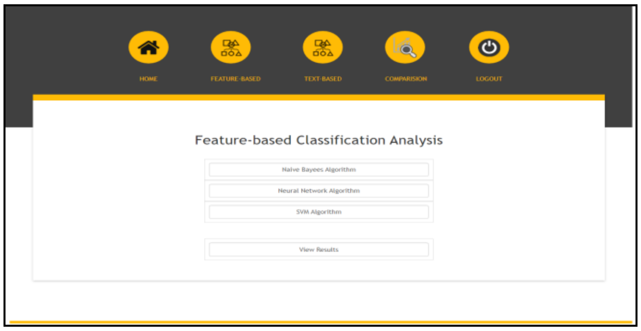


Figure:-2 Feature-based model page

Text-based Models

Compare to feature-based, the size of the text-data dataset is very high. The process time for building training model files is also high compare to feature-based model. Based on this reason, this system designed text-based model with two separate forms of training and testing. After training process, the system build the training models and save into physical formats of SAV (.sav) for avoiding the re-training process while prediction. As illustrated, three SAV files were created for three text-based classifiers of Naïve Bayes, SVM, and Neural Network.

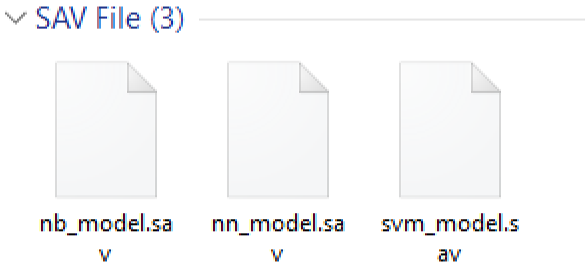
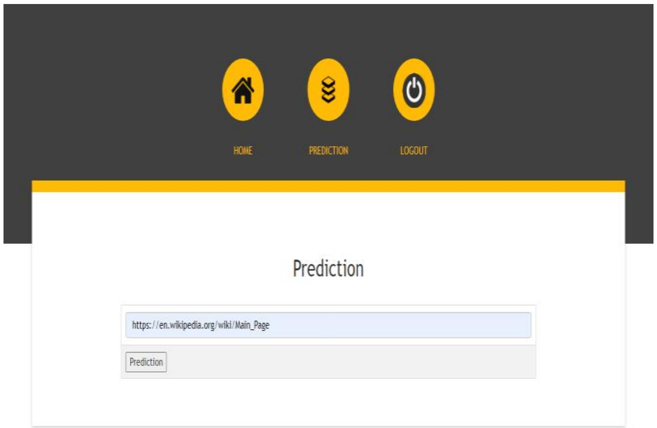


Figure:-3 Text-based models

Prediction page

The Prediction page can redirect from the main menu of the user portal, after verifying the user session, the system allows to access this page. As illustrated in , this page has inputs for entering the URL and initiating prediction with best model.



https://en.wikipedia.org/wiki/Main_Page

Figure:-4 Prediction Page

Classification Results

The classification results conducted for this system are covered in this section. This classification analysis depends on two types of data variants:

feature-based and text-based. This classification analysis used three ML algorithms for both models. Those are Neural Networks, SVM, and Naïve Bayes. This classification analysis depends on three aspects.

- The performance measure of ML models on the feature-based model.
- The performance measure of ML models on the text-based model.
- The performance measure of a feature-based and text-based model

The performance measure of ML models on the feature-based model

This classification analysis was conducted on the phishing website features dataset. This dataset is divided into two parts in a 70:30 ratio.70% of the Phishing Website Features dataset is taken for training and 30% of the dataset is used for testing. Based on the testing results, the accuracy score is calculated. The results of the accuracy scores of the ML models on the feature data are mentioned in Table 6.1 These accuracy results are graphically represented using a bar graph, illustrated in figure 6.12. In this classification analysis, I observed that the Neural Network algorithm got the highest accuracy score by comparing other ML models.

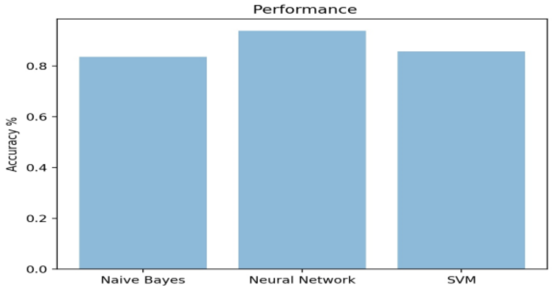


Figure:-5 Feature-based model accuracy scores

The performance measure of a feature-based and text-based model

This classification analysis was conducted on the phishing website text dataset. This dataset is divided into two parts in a 70:30 ratio.70% of the dataset is taken for training and 30% of the dataset is used for testing. Based on the testing results, the accuracy score is calculated. The results of the accuracy scores of the ML models on the text-based data are mentioned in Table these accuracy results are graphically represented using a bar graph, illustrated in figure 6.13. In this classification analysis also, the Neural Network algorithm got the highest accuracy score by comparing other ML models.

Table represents the accuracy results of the ML models in the text-based dataset.

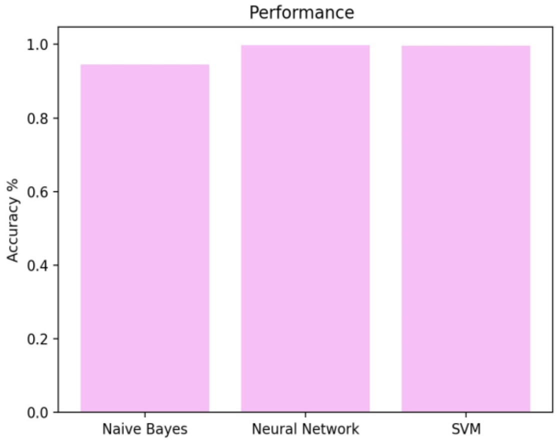


Figure:-6 Text-based model accuracy scores

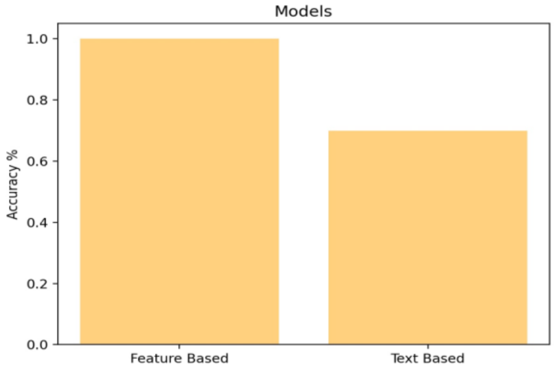


Figure: - 7 Accuracy Scores

4. CONCLUSIONS AND FUTURE WORKS

This project is focused on the build a detection system of bully data in social networks. To build a classification model this system has taken two datasets, namely, Hate Speech and Offensive

Language Dataset' and 'Harassment-Corpus Dataset'. For the prediction system, analyzed with the ML classifiers of Support Vector Machine (SVM), Naïve Bayes (NB), and Neural Network (NN) and implemented performance calculations to identify the best algorithm. In these results, the neural network algorithm got high accuracy, precision, recall, and f1-score comparative other algorithms. Based on the system results, the research questions answered and mentioned in the following, In future work suggesting to build a system to detect offensive content on media data such as images and videos.

5. REFERENCES

1. Alex Liddle, Dataset of Malicious and Benign Webpages, 2020, [Online] Available at: https://www.kaggle.com/code/alexliddle/semi-supervised-machine-learning-99-accuracy/data. Last Accessed 2nd, May, 2022.
2. Chaitanya B. 2020. Real python. [Online] Available at: https://realpython.com/python-mysql/ Last Accessed: 7th June, 2022.
3. Kunal Jain. 2015. Analytics vidhya. [Online] Available at: https://www.analyticsvidhya.com/blog/2015/01/scikit-learn-python-machine-learning-tool/ Last Accessed: 2nd July, 2022.
4. M. Zouina and B. Outtaj, (2017), "A novel lightweight URL phishing detection system using SVM and similarity index," Human-centric Computing and Information Sciences.
5. O. Sahingoz, E. Buber, O. Demir and B. Diri, (2019), "Machine learning based phishing detection from URLs," Expert Systems with Applications.
6. Pinto Ferreira, Ricardo & Martiniano, Andréa & Napolitano, Domingos Márcio & Romero, Marcio & Gatto, Dacyr & Farias, Edquel & Sassi, Renato. (2018). Artificial Neural Network for Websites Classification with Phishing Characteristics. Social Networking.
7. Rami Mustafa A Mohammad, Phishing Websites Data Set, 2015, [Online] Available at: https://archive.ics.uci.edu/ml/datasets/phishing+websites#. Last Accessed 1st April, 2022.
8. Rohit Dwivedi, Analytics Steps, 2020, [online], Available at: https://www.analyticssteps.com/blogs/what-naive-bayes-algorithm-machine-learning, Last Accessed: 3rd, July, 2022.
9. S., Adewole & Raheem, Muiz & AbdulRaheem, Muyideen & Oladipo, Idowu & Balogun, Abdullateef & Baker, Omotola. (2021). Malicious URLs detection using data streaming algorithms. Jurnal Teknologi dan Sistem Komputer.
10. Sun, Code Brust, 2020, [Online], Available At: https://codeburst.io/installing-and-configuring-mysql-with-django-a7b54b0f27ce. Last Accessed: 3rd June, 2022.
11. Sunil Ray, Analytics Vidhya, 2017, [online], Available at: https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/ Last Accessed: 3rd, July, 2022.