JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Developing an Identification-Driven Phishing Site Classifier

Prof. Ms. D.Vandana[1] , Duddyala Neeraja[2]

[1,2]G.Narayanamma Institute of Technology and Science,
[1,2] G.Narayanamma Institute of Technology and Science,
[1,2]Name of Department IT,
[1,2] Hyderabad, Telangana State, India.

Abstract : In today's digital world, there are a lot of websites that help individuals with different things including information sharing, marketing, socialising, and other things. However, some websites handle harmful or phishing operations. Phishing is the practise of gathering private data from the customers, including personal and financial details. Users may be threatened by this move. Phishing Websites are the online websites that conduct phishing operations. A common study topic and a requirement for browsers, online apps, and other applications is the detection of phishing sites. To identify phishing websites, several strategies have been researched that make use of various methodologies. Most of studies depended on the data mining and text mining techniques. Earlier researches have relied on text-based frameworks, which are constructed using text-data from web pages. This study concentrated on text-based and feature-based methods for classification analysis to identify the suitable model to predict the phishing websites. This study utilized Machine Learning algorithms, namely, Naïve Bayes, Support Vector Machine and Neural Networks algorithms. Based on comparative analysis, identify the suitable model for prediction of phishing websites.

***IndexTerms* - : Phishing Websites, Text-based ,Feature based,Naïve Bayes,SVM,Neural netwok.**

I. Introduction

For a variety of reasons, browsing many websites on a regular basis has become ingrained in the world of the internet. There are a tonne of trustworthy websites that support research, entertainment, and other activities. On the other hand, there are websites that expose consumers to security risks.[1] These websites impersonate reliable websites in order to obtain visitors' personal and business information. The identification of phishing websites is a crucial topic of research in computer security. Several various sorts of approaches were proposed in this study to detect phishing websites. Sahingoz et al. suggested a Phishing Webpage De-tection approach that incorporates NLP properties, such as word frequency, website domain comparison, word count mean, etc.[3] (2019)and classified text data along with website context data using machine learning techniques. Zouina et al. (2017) carried out phishing website recognition using website elements. To anticipate phishing sites, this predictive model used the Support Vector Machine's supervised machine learning technique.[4] Still, it only used six features from the website to make predictions. Text data is more frequently used to detect phishing sites. Text mining or machine learning techniques are used to forecast and train text data. Ade-wole et al. has presented a prediction model for the use of online text data in the identification of phishing websites. (2021).[5]

1.1 Proposed Model

This suggested model was created to identify the best model between feature-based and text-based approaches for phishing website prediction.[6] The main goal of this effort is to identify an appropriate classification model for phishing site detection. The best model for phishing site identification is found by comparing performance ratings across text-based and feature-based models using machine learning methods. [7]Three machine learning models—Naïve Bayes, Support Vector Machine, and Neural Network Algorithms—were utilised in the categorization study.[9]

1.2 Objectives

The main objectives of this proposed system of "Detection of Phishing sites" are mentioned in the following.

- Identify the datasets for both text-based and feature based model.
- Create classification models by using preprocessing strategies as necessary
- Detects the suitable model for detection of phishing sites.
- Implement prediction modules in a web-based application.

II.Literature Survey

Using an adaptable white list, Jain & Gupta (2016) suggested a phishing website detection mechanism that would alert users to the phishing activities. This strategy's definition is based on an examination of the website characteristics of both trustworthy and fraudulent websites. The information about the genuine websites is revised on the whitelist in accordance with the forecast findings. Based on a few factors, such as the quantity of internal links and the number of connections from within one's own domain, compare the value to the threshold.

Chrome plug-in was created with machine learning classification by Bohacik et al. (2020) to identify phishing sites. This method made use of eight features of the webpages, including @ symbol, proportion of other domain connections, IP address, and other things. The module applied 10-fold cross-validation and the C4.5 decision tree model to make the prediction.

Using text attributes and features of the URL, Aljofey et al. (2022) suggested a prediction method for phishing websites. The URL features data is depends on the text of the URL. To create tokens for the text characteristics, NLP methods were utilized to create webpage text data from the URLs. Phishing websites may be predicted by categorizing the data of URL attributes and text features. XGBoost, Random Forest, Naive Bayes, and Linear Regression used in classification analysis. The XGBoost algorithm received the highest performance in their findings.

Using text characteristics, Sahingoz et al. (2022) developed a method for predicting phishing sites. However, the text feature in this work is not solely reliant on token generators. Using the length of both long and short words, key phrases, brand word density, etc., this work retrieved text characteristics. Various machine learning algorithms were utilised for categorization analysis. The Random Forest algorithm received the greatest performance in their findings.

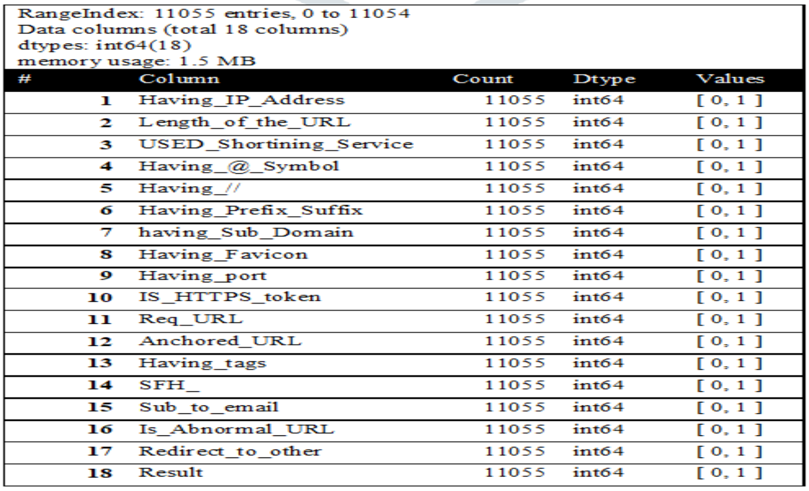
The WEKA technique was used by Geyik et al. (2021) to examine the identification of phishing sites using their URLs. In this study, URL data is viewed as having characteristics like the amount of reserved and unreserved characters, domain, length, etc. In a classification examination using a few different methods, the random forest approach did better.

III.Feature-based model classification analysis

Predictions regarding anything need analysis of the input data.[9] There are several formats in which the input data may be stored, such as text-based, feature-based, image-based, etc. Text-based and feature-based models were compared using this method.[10] This system considered the pertinent data or attributes of the web pages in its feature-based model. [11]The attributes of webpages, such as the length of the URL, the amount of symbols, HTTP or HTTPS token, etc. To meet the requirements of the feature-based classification analysis, a phishing features dataset was employed in this feature-based approach. (Rami, 2015).[12]

3.1 Dataset

The feature-based classification technique used the "Phishing Websites Dataset" (Rami, 2015) to predict phishing websites.[13] This dataset contains two labels, 1 and 0, and may be accessed as a CSV file. The feature-based dataset's description, created using Pandas Python API is shown in table 4.1.



```
RangeIndex: 11055 entries, 0 to 11054
Data columns (total 18 columns)
dtypes: int64(18)
memory usage: 1.5 MB
```

#	Column	Count	Dtype	Values
1	Having_IP_Address	11055	int64	[0, 1]
2	Length_of_the_URL	11055	int64	[0, 1]
3	USED_Shortining_Service	11055	int64	[0, 1]
4	Having_@_Symbol	11055	int64	[0, 1]
5	Having_//	11055	int64	[0, 1]
6	Having_Prefix_Suffix	11055	int64	[0, 1]
7	having_Sub_Domain	11055	int64	[0, 1]
8	Having_Favicon	11055	int64	[0, 1]
9	Having_port	11055	int64	[0, 1]
10	IS_HTTPS_token	11055	int64	[0, 1]
11	Req_URL	11055	int64	[0, 1]
12	Anchored_URL	11055	int64	[0, 1]
13	Having_tags	11055	int64	[0, 1]
14	SFH_	11055	int64	[0, 1]
15	Sub_to_email	11055	int64	[0, 1]
16	Is_Abnormal_URL	11055	int64	[0, 1]
17	Redirect_to_other	11055	int64	[0, 1]
18	Result	11055	int64	[0, 1]

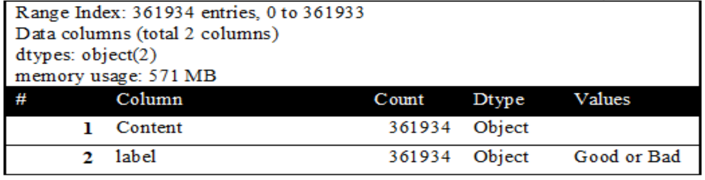
Table 3.1: Description of feature-based dataset

IV.Text-based model classification analysis

The website text or content of the online websites was considered under the text-based approach. [14]The content of the legitimate and phishing webpages was examined using the text-based dataset that contains the content of webpages labelled as genuine or phishing.(Alex, 2020).

4.1 Dataset

The text-based algorithm used the online data gathering to categorise and predict the difference between good and bad websites (phishing and genuine). (Alex 2020). This dataset, which is split into two categories—good and bad—is available as a CSV file.[15] The overview of the text-based dataset produced with using Pandas Python API is shown in table 4.1.



```
Range Index: 361934 entries, 0 to 361933
Data columns (total 2 columns)
dtypes: object(2)
memory usage: 571 MB
```

#	Column	Count	Dtype	Values
1	Content	361934	Object	
2	label	361934	Object	Good or Bad

Table4.1: Description of text-based dataset

V.Classification Models

For the feature-based and text-based models in the classification study of this system, three machine learning models were employed. [16]The two sections of the online content collection are divided in a 70:30 ratio. Of the dataset of internet material, thirty percent is used for testing and seventy percent is for training. The testing data for three machine learning algorithms are used in this study to determine the performance outcomes.[17] .

- Naïve Baye
- Support Vector Machine
- Neural Network

5.1 Naïve Bayes implementation for both Feature-based and Text-based

The Naive Bayes algorithm is a supervised learning technique that is based on the Bayes theorem [Rohit Dwivedi, 2020]. The Scikit-Learn Python Library was utilised in the implementation of this classification model. The Sklearn Library offers a number of Naive Bayes model variations. [17]In this work, the feature-based and text-based datasets were classified using the Multinomial Nave Bayes algorithm.

FEATUREBASED

```
Ifalgo=='nb';                                                    alg=MultinomialNB()
train_file="featuresDataset.csv"
df=pd.read_csv(train_file)
 y=np.array(df['result'])
x=np.array(df.drop(['result'],1)]
 alg=alg.fit(x,y)
```

TEXT BASED

```
 def nbclassfy(train_file='Webpages_Classification_test_data.csv', model='nb_model.sav'):
 train_data= pd.read_csv(train_file)
 tfidf = TfidfVectorizer(stop_words='english', use_idf=True, smooth_idf=True)  # TF-IDF
svm_pipeline = Pipeline([('lrgTF_IDF', tfidf), ('lrg_mn', MultinomialNB())])
filename = modelpickle.dump(svm_pipeline.fit(train_data ['content'].  train_data['label']), open(filename, 'wb'))
print("Model Successfully Trained")
```

5.2 Support Vector Machine implementation for both Feature-based and Text-based

FEATURE BASED

```
deftrain():
 alg=linearSVC()
 train_file="FeaturesDataset.csv"
 df=pd.read_csv(train_file)
 y=np.array(df['Result'])
 x=np.array(df.drop([Result'],1))
alg=alg.fit(x,y)
```

TEXT BASED

```
defsvmclassfy(train_file='Webpages_Classification_test_data.csv',
 model='svm_model.sav'):
train_news = pd.read_csv(train_file)
tfidf = TfidfVectorizer(stop_words='english', use_idf=True, smooth_idf=True)  # TF-IDF
svm_pipeline = Pipeline([('lrgTF_IDF', tfidf), ('lrg_mn', LinearSVC())])
filename=modelpickle.dump(svm_pipeline.fit(train_news['content'].val
ues.astype('U')[0:100000], train_news['label'][0:100000]), open(filename, 'wb'))
print("Model Successfully Trained")
```

5.3 Neural Networks implementation for both Feature-based and Text-based

```
FEATURE BASED
deftrain():
  alg=linMLPClassifier()
 train_file="FeaturesDataset.csv"
 df=pd.read_csv(train_file)
 y=np.array(df['Result'])
x=np.array(df.drop([Result'],1))
 alg=alg.fit(x,y)
 TEXT BASED
def nnclassfy(train_file='Webpages_Classification_test_data.csv', model='nn_model.sav'):
train_news = pd.read_csv(train_file)
 tfidf= TfidfVectorizer(stop_words='english', use_idf=True, smooth_idf=True)  # TF-IDF
svm_pipeline = Pipeline([('lrgTF_IDF', tfidf), ('lrg_mn', MultinomialNB())])
filename = model pickle.dump(svm_pipeline.fit(train_news['content'], train_news['label']), open(filename, 'wb'))
 print("Model Successfully Trained")
```

VI. Implementation Results

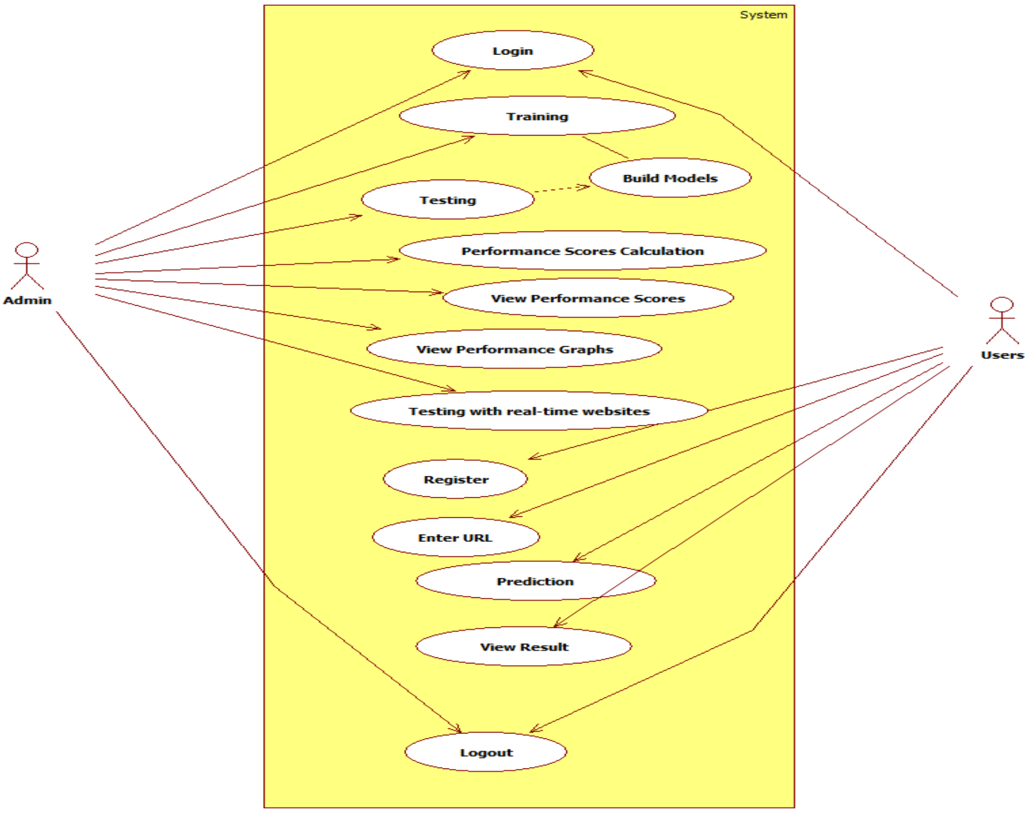
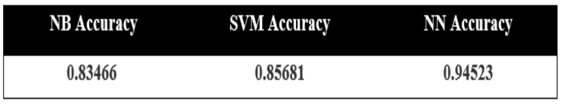


Figure 6:Use case diagram for phishing a website.

6.1 Feature-based classification analysis results

This module displays, in a bar graph style, the accuracy ratings for each machine learning method throughout the test dataset for the feature-based module.Figure 6.5 illustrates the findings, which indicate that the neural network method outperformed the other techniques.



NB Accuracy	SVM Accuracy	NN Accuracy
0.83466	0.85681	0.94523

Table 6.1: Accuracy scores

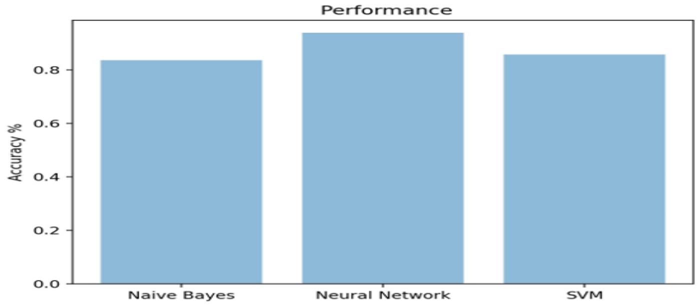
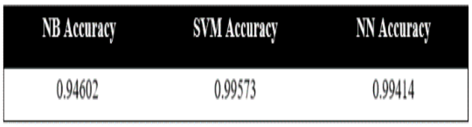


Figure 6.1: Accuracy scores of feature-based

6.2 Text-based classification analysis page

To meet the requirements of the text-based classification analysis, an online text dataset containing phishing sites was employed in this text-based approach. Three classifier implementations are available on this result page to get the model file creation process started for the machine learning algorithms. Additionally, this module uses model files for all machine learning methods to predict test dataset outcomes. This module displays, in a bar graph style, the accuracy ratings for each machine learning method across the test dataset findings for the text-based module.Figure 6.7 illustrates the findings, which indicate that the neural network method outperformed the other techniques.



NB Accuracy	SVM Accuracy	NN Accuracy
0.94602	0.99573	0.99414

Table 6.2: Accuracy scores

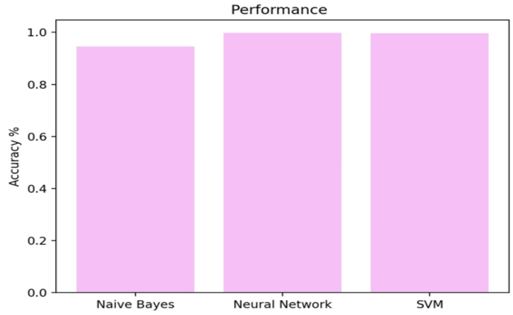


Figure 6.2: Accuracy scores of text-based

VII. Conclusions and Future Work

The act of phishing involves asking customers for sensitive information, which puts consumers at risk. It is a frequent study topic and a need for browsers, online apps, and other programmes to detect phishing sites. Text mining and data mining techniques were used in the majority of investigations. Previous research have employed text-based frameworks, which are constructed from text data from websites. This study aimed to determine the most effective model for phishing website prediction by analysing text-based and feature-based classification analysis techniques. Neural networks, support vector machines, and naive bayes were the machine learning techniques employed in this work. These methods were applied to two distinct text-based and feature-based datasets.The neural network approach produced improved performance results in terms of accuracy measure for both classification analyses. Accuracy to manually marked real-time webpages was also done by this approach. In comparison to the text-based neural network model, the feature-based neural network model performed better in this study.
In the future, advanced classification models like Convolutional Neural Networks (CNN), R-CNN, and others should be taken into account for the prediction of phishing websites.

VIII. 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, Oct, 2022.

[2] Aljofey, Ali & Jiang, Qingshan & Rasool, Abdur & Chen, Hui & Liu, Wenyin & Qu, Qiang & Wang, Yang. (2022). An effective detection approach for phishing websites using URL and HTML features. Scientific Reports. 12. 8842. 10.1038/s41598-022-10841-5.

[3] Bohacik, Jan & Skula, Ivan & Zábovský, Michal. (2020). Data Mining-Based Phishing Detection. 27-30. 10.15439/2020F140.

[4] Chaitanya B. 2020. Real python. [Online] Available at: https://realpython.com/python-mysql/ Last Accessed: 7th Oct, 2022.

[5] Geyik, Buket & Erensoy, Kubra & Koçyiğit, Emre. (2021). Detection of Phishing Websites from URLs by using Classification Techniques on WEKA. 120-125. 10.1109/ICICT50816.2021.9358642.

[6] Jain, Ankit & Gupta, B B. (2016). A novel approach to protect against phishing attacks at client side using auto-updated white-list. EURASIP Journal on Information Security. 2016. 10.1186/s13635-016-0034-3.

[7] Kunal Jain. 2015. Analytics vidhya. [Online] Available at: https://www.analyticsvidhya.com/blog/2015/01/scikit-learn-python-machine-learning-tool/ Last Accessed: 2nd Oct, 2022.

[8] M. Zouina and B. Outtaj, (2017), "A novel lightweight URL phishing detection system using SVM and similarity index," Human-centric Computing and Information Sciences.

[9] O. Sahingoz, E. Buber, O. Demir and B. Diri, (2019), "Machine learning based phishing detection from URLs," Expert Systems with Applications.

[10] 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.