2021 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES) | 978-1-6654-3521-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICSES52305.2021.9633889

Malicious URL Detection using NLP, Machine Learning and FLASK

A. Lakshmanarao[1], M.Raja Babu[1] and M M Bala Krishna[2]

[1]Department of Information Technology, Aditya Engineering College, Surampalem, A.P, India
[2]Department of Computer Science& Engineering, St Mary's Women's Engineering College, Guntur, A.P, India
E-mail : laxman1216@gmail.com raajababu.makineedi@gmail.com balakrishna2508@gmail.com

Abstract :- A URL created to attack with spam or fraud is known as a malicious/phishing URL. Viruses are downloaded into the system if the user clicks such URLs. Malicious URLs can lead to phishing and spam. With phishing, user credentials, valuable information is compromised. So, it is important to identify safe links and malicious links. Cyber-attacks are attempting with the origin of malicious URLs Phishers are manipulating their cyber attacking techniques rapidly. Machine Learning is a field of study where a system learns from previous experience and reacts to future events. Machine Learning methods are useful for resolving security applications. In this paper, authors proposed machine learning oriented solution for detecting malicious websites. For experiments, a Kaggle dataset with a large number of URLs (above 5,00000 URLs) is used. We applied three techniques for text feature extraction count vectorizer, hashing vectorizer-IDF vectorizer, and later build a phishing website detection model with four ML classifiers Logistic Regression, K-NN, Decision Tree, Random Forest.The ML model with hash vectorizer and random forest achieved 97.5% accuracy. We also created a web app using Flask for detecting the entered URL is malicious or not.

Keywords—Malicious URL, Kaggle, Machine Learning, Flask..

I. INTRODUCTION

In today's digital world, the risk of malicious URLs is very harmful. Users have no time to think about the URL before clicking on that. The user faces a lot of cybersecurity issues after clicking malicious URLs. A user may receive a malicious URL through email or WhatsApp. A malicious URL is also similar to a legitimate URL. So, it is difficult to identify which is malicious, which is not. Unfortunately, all the URLs use the same structure with 5 component structure, "subdomain", "top_level domain", "2nd-level-domain", "subdirectory". For example, consider a URL https://shop.yourstore.com/hats.For this URL, HTTPS is a scheme, the shop is the subdomain, your store is the second-level domain, com is top-level domain and hats is the subdirectory. If the malicious URL clicked, the consequences are un-known. A virus file may be downloaded to the system or user credentials can be stolen or it can be a backdoor malware.85% of the spam emails received in our mailbox are malicious links [1],[2]. There are several ways to detect malicious URLs.Some of the common techniques are secure email gateway, sandboxing, installing security protection plugins in the browser. But all the techniques had limitations too. In recent years,ML and DL methodologies are widely using for cybersecurity. Machine Learning able to read large volumes of data for analysis. There is a vast number of algorithms available for data analysis. Choosing a right algorithm is a crucial part of applying machine learning to cybersecurity. So, with the invention of machine Learning, cyber experts are having more choices for detecting phishing websites.

II. RELATED WORK

Applying machine learning and deep learning to cybersecurity was done from last several years. Several authors proposed ML/DL mechanisms for detecting malicious URLs. Saleem Raja[3] et.al used shallow learning techniques. They extracted lexical features of URLs and proposed ML classification models. With Random forest.they claimed best accuracy. For classification. Doyen Sahoo[4] et.al discussed literature reviews on the application of various machine learning techniques for the detection of malicious URLs. They also presented practical issues and open problems in malicious URL detection. Sandra Kumi [5] et.al applied the data mining approach CBA (Classification using Association) to detect malicious URLs and achieved an accuracy of 95.8%. Patil Bhavesh[6] et.al proposed a ML model with CSS data for phishing website detection and achieved good results. Frank Vanhoenshoven [7] et.al proposed RF and deep learning for malicious URL detection and achieved a good accuracy rate. They applied Naive Bayes, Multi-Layer Perceptron, Decision Trees, Support Vector Machines, Random Forest, and k-NN and achieved better results with random forest and multilayer perceptron. Clayton Johnson [8] et.al proposed a feature selection-based DL model for malicious URL(website) detection. In their work, they compared various algorithms like K-NN, CART, Random Forest, deep learning and achieved accuracy over 96% with random forest. Cho Do Xuan [9] et.al proposed URL classification with machine learning big data technologies. They selected few attributes by avoiding dataset complexity in terms of processing time and accuracy. Dharmaraj R. Patil[10] et.al proposed a feature-oriented multi-class classification model for

malicious website detection and claimed that they achieved good results.A. Lakshmanarao [11] et.al applied ranking algorithms to finalize a fusion approach and with that fusion approach achieved a good detection rate.

III. Proposed Methodology

The phishing URLs dataset is downloaded from Kaggle.Later text preprocessing techniques were applied to the collected dataset. After applying text pre-processing all the data is cleaned. Then various text processing NLP techniques are applied. We applied three NLP techniques namely TF-IDF vectorizer,count vectorizer(CV),hashing vectorizer(HV). We applied these three techniques separately. Later we applied various machine learning classifiers for the prediction of phishing URLs.The proposed framework was depeicted in Figure-1.

Dataset Preparation

The phishing urls dataset was collected from Kaggle [12]. The dataset contains 5,49,346 URLs, in which 1,56,422 urls are bad urls (phishing urls) and 3.92.924 urls are good URLs. The details of the dataset are shown in Table-1.

TABLE 1 DATASET INFORMATION

Source of dataset	No. of good urls	No. of bad urls	Total No.of urls
Kaggle	3,92,924	1,56,422	5,49,346

Text Preprocessing

Machine Learning and deep learning algorithms are capable of handling numeric data only. If the input data is other than numeric, first the data to be converted to numeric format. The only, ML or DL algorithm works on the data. Here the dataset contains URLs. It is text data, not numeric. So, first, this text data must be converted to numeric format. To convert into numeric, first, the given text data must be cleaned. For text cleaning, we used the python NLP package. We removed stopwords from the data. There is no use of stopwords for classification. Examples of stopwords are 'is','was','has' etc. Later we applied the stemming process. The stemming process removes prefixes in the words. For example, a word "playing" con be converted as play. We also removed unnecessary symbols in the text data.

Feature Extraction from Text

Machine Learning algorithms need input to be in number format. So, the cleaned URL text must be converted to a numeric format before applying machine learning algorithms. We applied three text encoding techniques namely Hashing Vectorizer. count vectorizer, TF-IDF.

Count Vectorization

Count Vectorizer is a simple technique for converting text to numeric format. It creates a column for every word in the dataset. If there are m rows, n different words in the dataset. Then count vectorizer makes the m*n vector. The count of each word is placed in the [i,j] position of the vector(Here [i.j] represents the ith row and jth column).

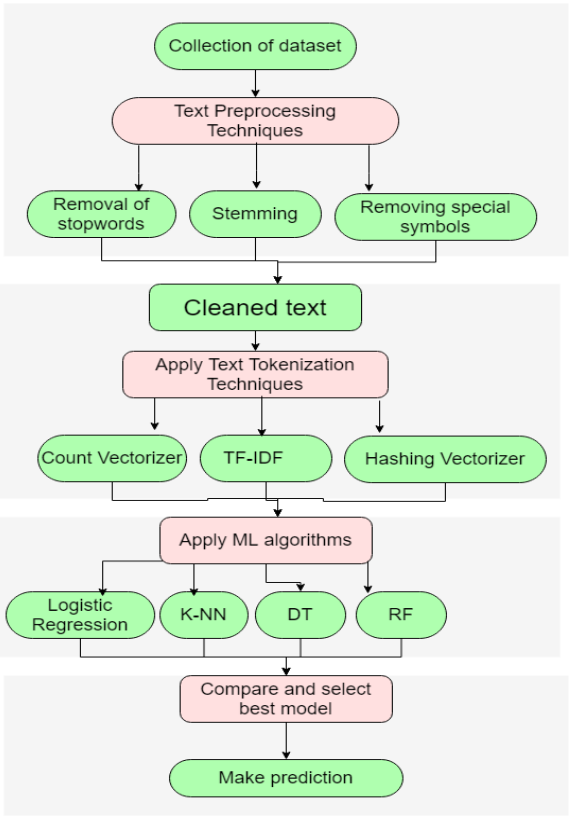

Fig. 1 Proposed Framework for malicious URL detection

TF-IDF Vectorization

Count Vectorizer simply maintains word counts. Although a word count is a good criterion for vectorization, it may not be helpful always.Sometimes,there are limitations with count vectorization.

For example, a word “is” appearing many times, but it may not be helpful for class label division. TF-IDF vectorizer considers word frequencies also. It uses two measures TF (Term Frequency), IDF (Inverse Domain Frequency). Term Frequency gives in-formation about how often a particular word appears in the input.IDF gives in-formation about the words which appear most of the times across documents.

TF (word, document) =log (1+f (word, document)

Here f (word, document) is the frequency of word w in document d

IDF (word, Documents) =log (N/f (word, Dataset)

Here N is number of documents in whole dataset (word, documents) is the frequency of word w in entire dataset.

Hashing Vectorization

If the vocabulary is large, it may be difficult for count and frequency criterions. Because vocabulary is large size f vector becomes large leading to more memory requirements. Hashing vectorizer uses one way hash of words in the dataset for converting words into integers. With this technique, the length of vector size can be arbitrary.

Machine Learning Algorithms

After encoding input text into a numeric format, we implemented several machine learning classifiers applied logistic regression, Decision tree, random forest, and K-NN classifiers for malicious URL detection. We applied all these four algorithms with three text encoding techniques. So, a total of 12 experiments is conducted for phishing URL detection. For comparison of these algorithms, we used accuracy measure.

IV. Experimentation and Results

Any supervised learning needs data to be divided into training -set and test-set. The algorithm is trained on a training set and later it creates a model. The generated model was used for testing the dataset. We divided the dataset into training and testing sets with a split ratio of 80%,20%. The details of splitting are shown in Table-2.

TABLE 2 DATASET SPLIT DETAILS

Total no. of samples	No. of Training samples	No.of Testing samples
5,49,347	4,39,477	1.09,870

Experimentation with Count Vectorizer

Four machine learning algorithms were applied with the encoding technique as a count vectorizer. The experimentation results are displayed in Table-3.

TABLE 3 RESULTS WITH COUNT VECTORIZER

Algorithm	Accuracy(%)
K-NN	89
DT	96.8
RF	97.1
Logistic Regression	96.5

Experimentation with TF-IDF

Four machine learning algorithms (Logistic Regression, RF, DT, K-NN) applied with another encoding technique TF-IDF vectorizer & the results are tabulated in Table-4.

TABLE 4 RESULTS WITH TF-IDF

Algorithm	Accuracy (%)
K-NN	60.9
DT	96.4
RF	97.4
Logistic Regression	96.1

Experimentation with Hashing Vectorizer

Four machine learning algorithms (Logistic regressing, RF, DT, K-NN) applied with another encoding technique TF-IDF vectorizer & the results are tabulated in Table-5. Random Forest given best accuracy of 97.5% with this vectorizer.

TABLE 5 RESULTS WITH HASHING VECTORIZER

Algorithm	Accuracy (%)
K-NN	87
DT	96.4
RF	97.5
Logistic Regression	95.3

Analyzing Results

After conducting all the 12 experiments, the accuracy comparison of three en-coding techniques is shown in fig-2 and Table-6.From figure-2.From figure-2,it is observed that all three text encoding techniques given good accuracy rates.With Count Vectroizer,RF given highest accuracy rate of 97.1%.With TF-IDF,random forest given high accuracy score of 97.4%.With Hashing Vectorizer,Random Forest given high accuracy of 97.5%.So,Hashing vectorizer and RF is best model.

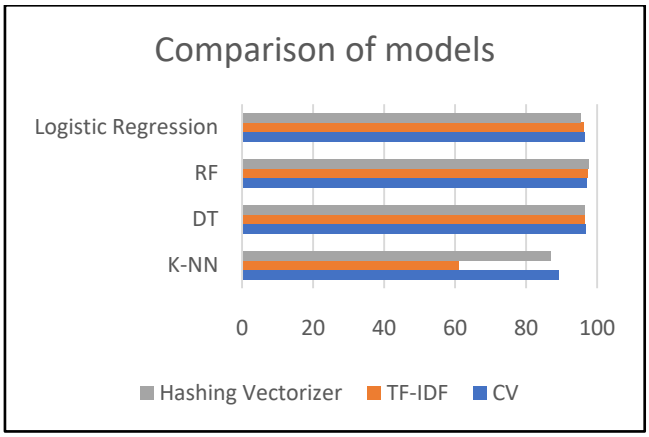

Fig. 2 Accuracy Comparison

TABLE 6 RESULTS OF ALL EXPERIMENTS

Algorithm	CV	TF-IDF	HV
K-NN	89%	60.9%	87%
DT	96.8%	96.4%	96.4%
RF	97.1%	97.4%	97.5%
LR	96.5%	96.1%	95.3%

Comparison woth previous work

In [15],they achieved a maximum accuracy of 95%.In this paper,the authors done experiments with various encoding techniques and achieved an accuracy of 97.5%.The comparison was shown in figure-3.

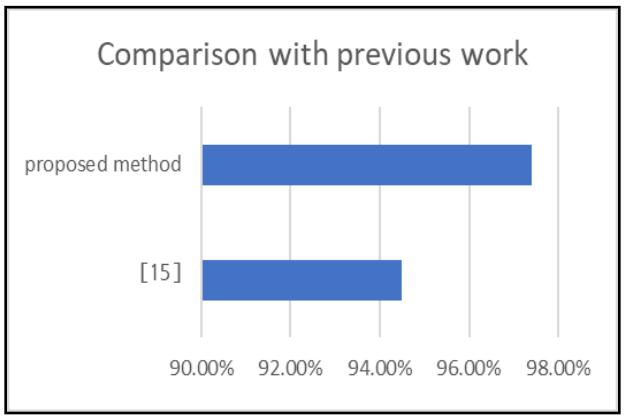

Fig. 3 Accuracy Comparison with previous work

Malicious URL Detector WebApp

After finalizing the encoding technique and machine learning algorithm, we build a web app for malicious URL Detector. We used python Flask for this purpose. Python 'pickle' is used to back up the models. We created pickle files for 'hashing vectorizer, 'Random Forest model'. We used these files for making predictions. The user enters the URL in the space provided. Then, user needs to click on "Predict" button. The user gets a response as "Malicious URL" or "Not a Malicious URL' based on the input given. It is depicted in figures 3.

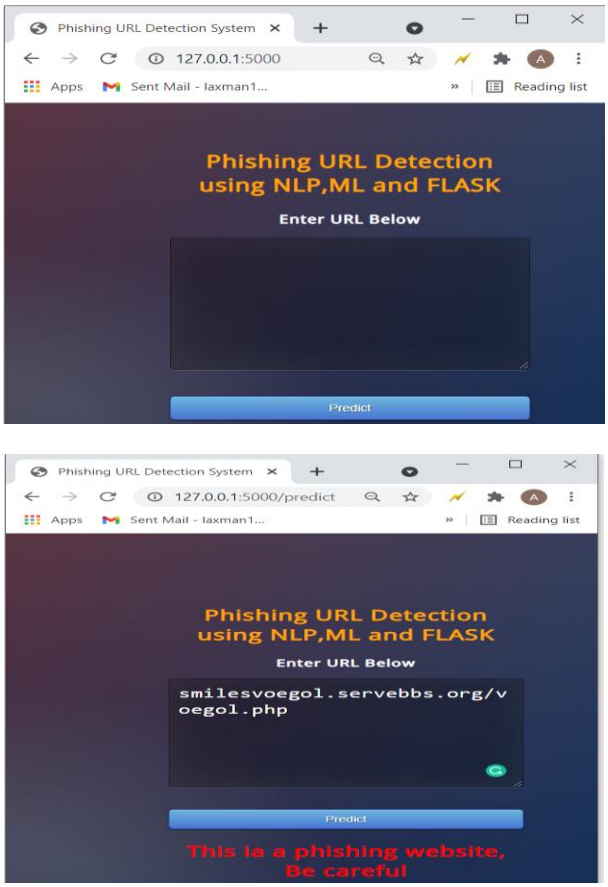

Fig. 4 Detecting Phishing URL using webapp

V. Conclusion

In this paper, we collected a phishing URLs dataset from Kaggle. We applied various text preprocessing techniques like removing stop words, stemming. Then, the authors applied three different text encoding techniques TF-IDF vectorizer, Count Vectorizer, Hashing Vectorizer. With these three techniques, We applied four shallow ML algorithms separately. Out of 12 experiments conducted, the proposed model achieved an accuracy rate of above 97% with Machine Learning random forest algorithm model with a hashing vectorizer encoding Technique. Later, we build a web app with a count vectorizer and random forest combination for malicious URL detection.Our model outper-forms previous models with 97.5% accuracy.

References

[1] https://www.zdnet.com/article/most-malspam-contains-a-malicious-url-these-days-not-file-attachments/

[2] Sheldon Williamson, K Vijayakumar (2021), Artificial intelligence techniques for industrial automation and smart systems, Concurrent Engineering, Volume 29, issue 3, pp 291-292.

[3] Saleem Raja,R.Vinodini,A.Kavitha, ":Lexical features based malicious URL detection using machine learning techniques",materialtoday Proceedings,Science Direct, April 2021. https://doi.org/10.1016/j.matpr.2021.04.041.

[4] D Sahoo, Ch Liu,Teven C.H.Hoi, "Malicious URL Detection using Machine Learning: A Survey", Vol. 1, No. 1,21 Aug 2019, arXiv:1701.07179v3 [cs.LG].

[5] K.Sandra,Lim ChaeHo,Sang-Gon Lee, "Malicious URL Detection Based on Associative Classification", entropy, MDPI, https://doi.org/10.3390/e23020182,2021.

[6] Bhavesh P.,Pranjal P. Omkar P.,Ketan T.,Kumbharkar P.B, "Malicious URL Detection Using Machine Learning",International Journal of Grid and Distributed Computing, Volume-13, pp. 2464– 2468,2020..

[7] F.Vanhoenshoven et.al , "Detecting Malicious URLs using Machine Learning Techniques", IEEE SSCI(Symposium Series on Computational Intelligence) ,978-1-5090-4240-1/16/$31.00 ©2016 IEEE.

[8] C Johnson,B Khadka,B Ram,Basnet,Doleck, "Towards Detecting and Classifying Malicious URLs Using Deep Learning",JOWUA(Journal of Wireless Mobile Networks), Ubiquitous Computing, and Dependable Applications ,11(4):31-48, Dec. 2020.

[9] Cho Do Xuan,H.D.Nguyen,Kumar,T.V.Nikolaevich, "Malicious URL Detection based on Machine Learning" International Journal of Advanced Computer Science and Applications,Vol. 11, No. 1, 2020.

[10] Dharmaraj R P., Jayantrao B P, "Feature-based Malicious URL and Attack Type Detection Using Multi-class Classification", The ISC Int'l Journal of Information SecurityJuly 2018, Volume 10, Number 2-pages- 141(162).

[11] A. Lakshmanrao, P. S. P. Rao,B.Krishna.,"Phishing website detection using novel machine learning fusion approach", ICAIS-2021(IEEE conference), pp. 1164-1169.

[12] K Vijayakumar, (2021), Computational intelligence, machine learning techniques, and IOT, Concurrent Engineering, Volume 29, issue 1, pp 3-5.