A Machine Learning approach for Detecting Malicious URL using different algorithms and NLP techniques

M.A. Waheed
Department of Computer Science and Engineering,
VTU, CPGS, Kalaburagi,
Karnataka, India
dr.mawaheed@gmail.com

Baswaraj Gadgay
Regional director
VTU Regional office
Kalaburagi,
Karnataka, India
baswaraj_gadgay@vtu.ac.in

Shubhangi DC
Department of Computer Science and Engineering,
VTU, CPGS, Kalaburagi,
Karnataka, India
drshubhangipatil1972@gmail.com

Vishwanath P
Department of Electronics and Communication Engineering
H.K.E. Society's S.L.N College of Engineering, Raichur
Karnataka, India
vishalpetli73@gmail.com

Qurat Ul Ain
Department of Computer Science and Engineering,
VTU, CPGS, Kalaburagi,
Karnataka, India
qurat1725@gmail.com

2022 IEEE North Karnataka Subsection Flagship International Conference (NKCon) | 978-1-6654-5342-4/22/$31.00 ©2022 IEEE | DOI: 10.1109/NKCON56289.2022.10126798

***Abstract*—A malicious URL is one that was made specifically to attack through spam or fraud. Due to the billions of dollars that are compromised, malicious URLs pose a severe threat to security software. Finding secure and phishing links is therefore crucial. Therefore, machine learning is quite helpful for resolving security-related challenges. In this study, we use about 5 lakh URLs that were retrieved from the Kaggle dataset. We are utilizing three NLP approaches, including the count vectorizer, hash vectorizer, and TF-IDF vectorizer. Six machine learning classifiers, including the decision tree, random forest, K-NN, NB, SVM, and logistic regression, were used in conjunction with all these techniques. The highest accuracy results of 98.2 percent are produced by random forest. To determine whether or not the URL supplied is malicious, we built a web app using Flask**.

***Keywords*— count vectorizer, Flask, Kaggle, Malicious URL, ML**

I. Introduction

The advancement of the World Wide Web (www) has made it easier for cybercriminals to attack with spam or fraud, which compromises the user's valuable information without their knowledge. If a user opens a malicious URL, harmful content will be installed immediately on their system. As users, we do not have time to check whether the URL clicked is safe or phishing. So, it is very difficult to distinguish between safe and malicious URLs. 85 percent of the spam emails we get contain malicious links. The reason for the increase in malware attacks is that there is no proper tool for checking for malicious links. So, the user must update their valuable information in a timely manner. Machine learning and deep learning play a significant role in resolving security challenges. The majority of phishing attacks share a number of features, making machine learning the most effective method for identifying them [11]. The study of machine learning gives the system the ability to act on future occurrences and learn from the past. For data analysis, ML can read huge amounts of data. There are several algorithms available for data analysis. So, it is very important to select the correct algorithm for applying machine learning to cybersecurity. Detecting malicious websites becomes much easier with the introduction of machine learning.

II. Related Work

It shows a lighter technique that merely considers the URL's lexical features. This study's findings demonstrate that the Random Forest model is more accurate than others [3].

A malicious URL is a serious cybersecurity threat. As well as scientific machine learning scientists, as well as experts and professionals in the cyber security business, this survey will assist them in better grasping the state of the art and supporting their own study. Additionally, open research difficulties and promising new paths for investigation were presented [4].

In order to identify malicious URLs, a data mining approach known as CBA is described. This method employs a training set of URLs as history data to uncover association rules in order to produce an appropriate classifier [5].

Depending on our suggested URL behaviours and features of a dataset, a machine learning strategy for identifying malicious URLs is described in this paper. Additionally, the use of big data technology has enhanced the ability to identify dangerous URLs based on unusual behaviour. The experimental data demonstrate that the specified URL properties and behaviour can significantly increase the potential to recognise malicious URLs. Accordingly, the provided technique might be considered an efficient and user-friendly method of detecting harmful URLs [9].

Machine learning classifiers have been used with two target methods to detect malicious URLs. Based on the target algorithm, a fusion classifier is chosen. Results with good accuracy are obtained using a fusion classifier with machine learning methods [12].

III. Proposed Model

A dataset from Kaggle containing URLs is used in the proposed framework. The obtained dataset was cleaned using text pre-processing methods. Additional NLP approaches, including a count vectorizer, a hash vectorizer, and a TF-IDF vectorizer, were used. Next, a machine learning classifier is used to predict the URL.

A. Methodology

1) Dataset preperation

Kaggle was used to collect the phishing URL dataset [12]. There are 4,50,176 URLs in the dataset, with 1,03,540 bad URLs and 3,46,636 good URLs.

2) Text Pre-processing

The dataset used, which includes URLs, was collected from Kaggle. Only text data is provided in the URL. Deep learning and machine learning algorithms only function with numerical data. The dataset must therefore be transformed into numeric form. The data from the dataset is converted, and then, using Python NLP packages, it is cleaned. Prefixes, stop words, and any unnecessary symbols were eliminated from the text.

3) Feature Extraction from Text

The URL text data needs to be in numbers prior to using ML techniques. Three NLP approaches, including count vectorizer, TF-IDF vectorizer, and hash vectorizer, were used.

a) Count Vectorization

It is an approach for decomposing a text into words before converting it to numbers. A matrix of the type m*n is created if there are m rows and n distinct words. It emphasises word count. Word counts are each saved in the [i,j] vector.

b) TF-IDF Vectorization

It is divided into two sections:

TF

Term Frequency, abbreviated as TF, measures how frequently a word appears in the input text. The calculation formula is as follows in equation 1:

$$TF = \frac{Word\ frequency\ in\ a\ document}{Total\ number\ of\ words\ in\ a\ document} \qquad (1)$$

This number is fixed at 1. It determines how frequently each word appears in proportion to all other words in a document.

IDF

IDF stands for "inverse document frequency," and it gives each word in a document a priority. Expressing equation 2 in log form as

$$IDF = \log(\frac{Total\ \ documents}{Documents\ containing\ word\ W}) \qquad (2)$$

The end outcome is the product of TF-IDF as follows in equation 3:

$$TF - IDF = TF * IDF \qquad (3)$$

c) Hashing Vectorization

The hashing approach is used to perform vectorization. Instead of locating the indices in the array, the hash function applies to the features and utilizes the hash values as indexes. Hashing is a good option since it immediately limits the size of the vector if the dataset is too big.

B. Machine Learning Algorithms

We used a variety of machine learning classifiers to detect malicious URLs after converting the input text to a numeric format, including decision trees, random forests, K-NN, SVM, logistic regression and classifiers. Three text encoding techniques were combined with all six algorithms. To compare these algorithms, we used an accuracy metric.

The following is the description of ML algorithms

1) Logistic regression

In this statistical analytic approach, observed data from a data collection are used to forecast a binary result, like no or yes. Its model makes predictions about a dependent variable by analyzing the correlation between one or more independent variables that are already present.

2) KNN

K-Nearest Neighbor is referred to as KNN. It is an algorithm for supervised machine learning. The technique can be used to solve classification and regression issues. A novel known factor must be either predicted or categorized based on its K nearest neighbors.

3) Decision Tree

A decision tree has a higher likelihood of overfitting. Its forecast performance for a dataset is poor when compared to that of other machine learning techniques. Information gain in a decision tree containing category factors results in a biased response for qualities with a larger number of categories.

4) Random Forest

The supervised learning approach uses the Random Forest algorithm, which is well-known in the machine learning community. Use it for either regression or classification in machine learning applications. To handle complicated problems and enhance the model's performance, it employs the notion of supervised methods that include integrating many classifiers.

5) Support Vector Machines

The supervised learning model known as SVM has lately gained popularity in the field of machine learning and pattern recognition. The "margin maximisation idea" serves as their foundation. They undertake structural risk reduction, which increases the classifier's complexity with the goal of getting great generalisation performance.

6) Naïve Bayes

The Bayes theorem is the core of naive Bayes classifiers, which are simply probabilistic classifiers. Task classification is carried out using the Naive Bayes algorithm. The fundamental idea behind classifiers is that every pair of features is independent of one another. The naïve bayes classification is expressed in equation 4 as

$$P(A|B) = P(B|A) * \frac{P(A)}{P(B)} \qquad (4)$$

C. System Architecture

The dataset was gathered through Kaggle. First, we read the gathered dataset from the.csv file, which contains all the URL's that contain legitimate and phishing URLs and saves them in the form of rows and columns. The dataset's null content was then removed using the feature extraction approaches (CV, HV, and TF-IDF). The data has been cleaned by that point. After the data had been cleansed, ML algorithms (LR, DT, K-NN, RF, and SVM, NB) were used. The best model is then chosen from this group by comparing all the models. The chosen model will then produce the optimal outcome.

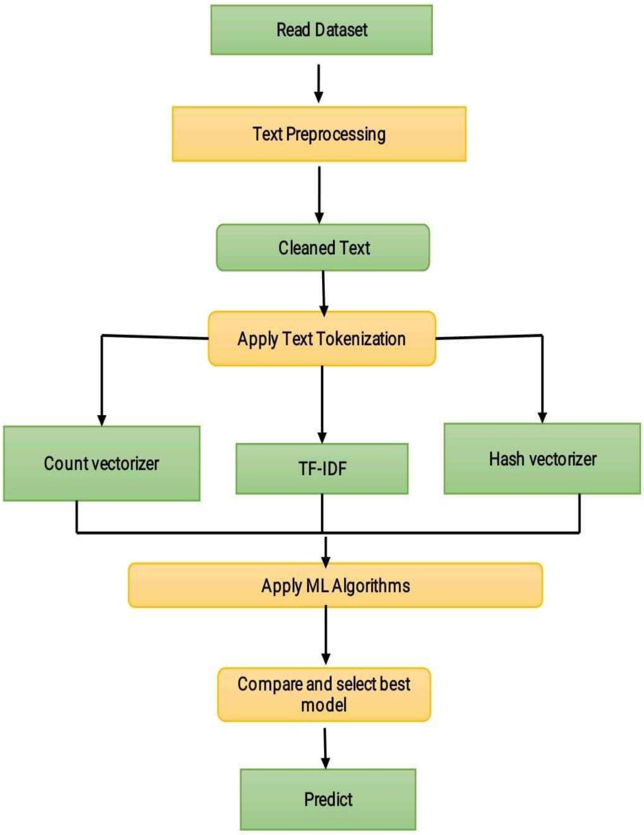
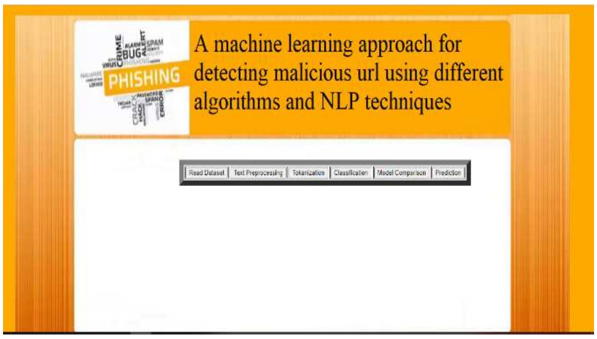


Fig.1 System Architecture

IV. RESULTS AND DISCUSSION

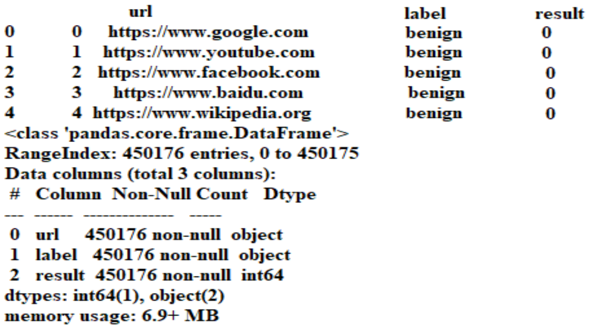
The proposed method makes use of the Kaggle dataset, in which 77 and 23 percent of the URLs, respectively, are benign and malicious. Six machine learning algorithms (DT, RF, LR, K-NN, and SVM, NB) were combined with three techniques (CV, HV, TF-IDF). Random forest is the most accurate algorithm model for producing outcomes with the highest degree of accuracy since it provides the highest accuracy among these algorithms.



A machine learning approach for detecting malicious url using different algorithms and NLP techniques

Fig.2 Main

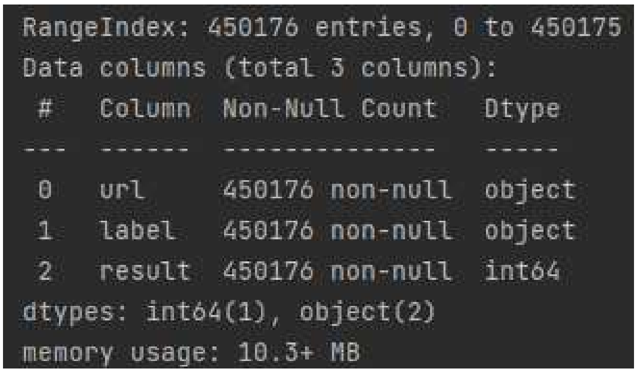
Fig.2 main shows the steps that are involved in proposed system used for detecting the malicious or benign URL.



```
                  url                                label        result
0         0   https://www.google.com                 benign         0
1         1   https://www.youtube.com                benign         0
2         2  https://www.facebook.com                benign         0
3         3     https://www.baidu.com                benign         0
4         4  https://www.wikipedia.org               benign         0
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 450176 entries, 0 to 450175
Data columns (total 3 columns):
 #  Column  Non-Null Count  Dtype
--- ------  --------------  -----
 0  url     450176 non-null  object
 1  label   450176 non-null  object
 2  result  450176 non-null  int64
dtypes: int64(1), object(2)
memory usage: 6.9+ MB
```

Fig.3 Read Dataset

An URL.csv file containing real and phishing URLs is used to store the data in Fig.3. The.csv file is a basic file format for storing and displaying data in the form of rows and columns.



```
RangeIndex: 450176 entries, 0 to 450175
Data columns (total 3 columns):
 #   Column  Non-Null Count   Dtype
---  ------  --------------   -----
 0   url     450176 non-null  object
 1   label   450176 non-null  object
 2   result  450176 non-null  int64
dtypes: int64(1), object(2)
memory usage: 10.3+ MB
```

Fig.4 Pre-processing

Fig.4 shows how the preprocessing technique is used to eliminate the null/empty content from the dataset file. After the data has been cleansed, it will be fed into our model to ensure that it is as accurate as possible.

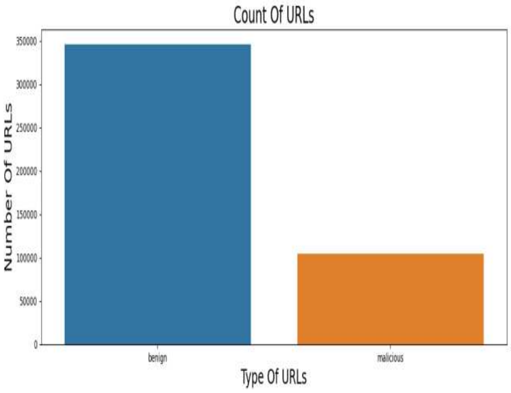


Fig.5 Count of URL's

Fig.5 shows the count of genuine and malicious URL present in the total number of URL present in the dataset.

TABLE I. Decision Tree, Random Forest, Logistic Regression, KNN, SVM and NB algorithm accuracy comparison Table

Algorithm	***Accuracy***
Decision Tree	0.975446237036849
Random Forest	0.9824535014787829
Logistic Regression	0.9761792818065269
K- NN	0.9761692818065269
SVM	0.9676555818065269
NB	0.951692818065269

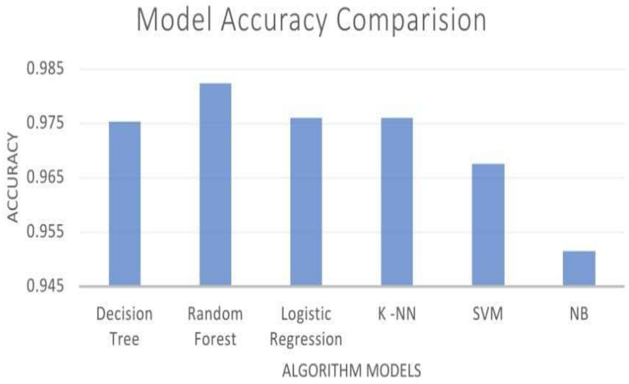
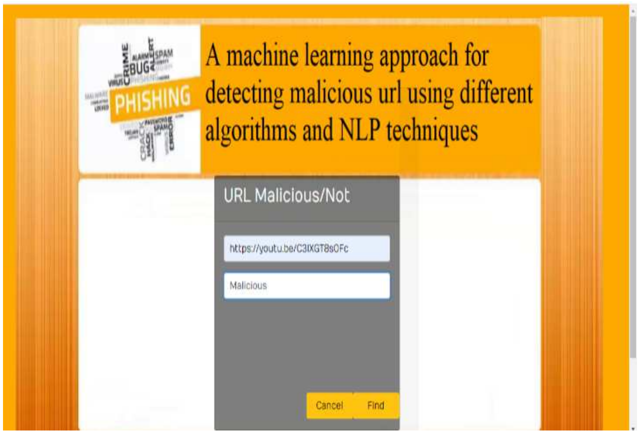


Fig.6 Model Accuracy Comparison Graph

Fig.6 and TABLE I. shows the accuracy of the ML algorithms in which Random Forest shows the highest an accuracy rate. So, it is considered as the best choice which provides the optimal results.


A machine learning approach for detecting malicious url using different algorithms and NLP techniques

URL Malicious/Not

https://youtu.be/C3IXGT8sOFc

Malicious

Fig.7 Finding Malicious/Not

Fig.7 shows a web application for identifying malicious URL's after perfecting the feature extraction methods and machine learning algorithm. For this, we employed Python Flask. When the user types the link into the box given. The user must then click the "Find" button. Depending on the information provided, the user determines if the response is a malicious URL or not.

V. Conclusion and Future Scope

Malicious URL/websites lead to spam. Cybersecurity is the major threat, so machine learning is used to resolve this issue. The proposed system uses machine learning approaches which helps in handling the security applications. In this paper, the three feature extracting techniques were used with the six machine learning algorithms. Out of the six machine learning algorithms, random forest provides the highest accuracy rate of 98.2%. The future can be utilized with network-based features and dynamic online content to find malicious information on web pages. Also, we have the option of using online learning in place of using collected data.

References

[1]. Gerardo Canfora and Corrado Aaron Visaggio. 2016. A set of features to detect web security threats. Journal of Computer Virology and Hacking Techniques (2016).

[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", material today Proceedings, Science Direct,April2021. 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) ,978-1-5090-42401/16/$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 Security July 2018, Volume 10, Number 2-pages- 141(162).

[11]. Daiki Chiba, Kazuhiro Tobe, Tatsuya Mori, and Shigeki Goto. 2012. Detecting malicious websites by learning ip address features. In Applications and the Internet (SAINT), 2012 IEEE/IPSJ 12th International Symposium on. IEEE.

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

[13]. K Vijayakumar, (2021), Computational intelligence, machine learning techniques, and IOT, Concurrent Engineering, Volume 29, issue 1, pp 3-5. [14] Neda Abdelhamid, Aladdin Ayesh, and Fadi Thabtah. 2014. Phishing detection based associative classification data mining. Expert Systems with Applications (2014).

[14]. Mr A Sankaran, S. Mathiyazhagan, Prasanth, M. Dharmaraj Detection Of Malicious Urls Using Machine Learning Techniques, International Journal of Aquatic Science ISSN: 2008-8019 Vol 12, Issue 03, 2021

[15]. .Hung Le, Quang Pham, Doyen Sahoo, and Steven CH Hoi. 2018. URLNet: Learning a URL Representation with Deep Learning for Malicious URL Detection. arXiv preprint arXiv:1802.03162 (2018).

[16]. Immadisetti Naga Venkata Durga Naveen, Manamohana K, Rohit Verma, “Detection of Malicious URLs using Machine Learning Techniques”, International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-8 Issue-4S2 March, 2019

.