JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 2, 2022, p. 2768 - 2775
https://publishoa.com
ISSN: 1309-3452

A Machine Learning Approach for Identifying and Detecting Hazards

Mr B Srinivas, CSE:BVCE Email:sriv.vasv@gmail.com
Dr Gunamani Jena, Prof CSE:BVCE Email:drgjena@gmail.com
Anguluri Charukesh, CSE:BVCE
Annamdevula Sai Anantha Lakshmi, CSE:BVCE
Dangeti Satyanarayana Prasad, CSE:BVCE
Maddisetti D Naga Lakshmi, CSE:BVCE

ABSTRACT

Every day, we log on to the internet and use it to conduct our business. As a result, browser vendors compete for users' attention by adding new features and enhancing existing ones, which exposes websites to hackers' attacks. Surfers, on the other hand, are looking for a rapid and precise model that can tell the difference between safe and harmful websites. In this study, we develop a new classification approach to evaluate and detect dangerous online sites using Machine Learning classifiers such as Random Forest, Support Vector Machine, Naive Bayes, Logistic Regression, and a specific URL (Uniform Resource Locator). Experimental data shows that the random forest classifier is more accurate than other machine learning classifiers, with an accuracy rate of 95%.

Keywords- hazard detection, hackers, internet surfing.

INTRODUCTION:

Internet banking, online shopping, bill payment, e-learning, and other services are becoming more widely available to consumers as the web expands rapidly, and individuals are using browsers [4] or web applications to access the web. Personal and sensitive information is at risk as the capabilities and functionalities of browsers continue to grow [3]. With just a single click on a rogue online site, unskilled visitors are easily taken advantage of by attackers, who can then exploit the page's vulnerabilities to gain remote access to the victim's web page. Since the internet is constantly expanding, it's imperative that web pages be accurately identified. Despite their shortcomings, blacklisting services were incorporated into browsers to address these issues[3]. On the subject of web page classification, this essay examines a self-learning approach to the task. We use four machine learning classifiers to split the internet site into two categories: benign and dangerous web pages.

Identify Malicious Websites on the Internet (1.1)

The internet's security is threatened by malicious web pages. It is common for rogue web pages to use drive-by download assaults to gain complete control of a user's machine. Malware executables can be downloaded and installed with just a single visit to a rogue website. For the first time, we are able to identify online sites that are either dangerous or harmless by using supervised machine learning.

1.2 Malicious content on websites

Access to websites with dangerous content is a risk on the Internet because they can serve as entry points for criminality or as a means for downloading things that can harm organisations, people, and the environment. Furthermore, website attack logs have been included in cyber attack reports in recent years; this information covers attacks perpetrated by current hazards discovered in new technologies, such as the Internet of Things. Due to the complexity of computer security, researchers have been experimenting with using machine learning algorithms to detect dangerous web content. In order to classify a website, this paper examines the implementation of a data analysis method using a framework that incorporates dynamic, static analysis, updated websites, and a low interaction client honeypot. Furthermore, it assesses the capability of four machine learning classifications based on the data examined.

1.3 Motivation:

As the World Wide Web continues to grow, we are confronting an increasing threat from bad web pages such as phishing, malware, and spamming. There are several flaws in the work of detecting dangerous web pages and determining their threat categories.

1.4 Problem Description:

With just a single click on a rogue online site, unskilled visitors are easily taken advantage of by attackers, who can then exploit the page's vulnerabilities to gain remote access to the victim's web page. Since the internet is constantly

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 2, 2022, p. 2768 - 2775
https://publishoa.com
ISSN: 1309-3452

expanding, it's imperative that web pages be accurately identified.To meet the issues, blacklisting services were included in browsers, although they had various drawbacks, such as incorrect listing.

1.5 Goal:

The proposed detection method employs machine learning methods and relies on these URL-based attributes to identify dangerous from benign web pages.

2.1 Aims and objectives of the project:

An unwitting user is infected with malware that takes personal information, sends her to dangerous sites, or compromises her computer so that more attacks can be conducted.. Filtering wild web pages, capturing a wide range of malicious characteristics, systematically combining features, semantic implications of feature values on characterizing web pages, and the ease and cost of flexibility and scalability of an application are some of the remaining open issues in the detection of malicious websites, despite the promise of existing approaches.

2.2 Research Contribution

As part of our approach, we address questions about current methodology, web page features, and analysis/detection strategies, as well as the broader problem of detecting fraudulent websites. One of the key objectives of our proposal is to uncover new and less resource-consuming harmful websites in the wild.

Methodologies for 3 Proposals

Random forest and support vector machine (SVM) classifiers are used to construct a new method for assessing and classifying dangerous online sites. Naive Bayes and logistic regression are used to train the classifiers to detect malicious web pages based on variables retrieved from the URL (Uniform Resource Locator).

Logistic Regression, Random Forest, Decision Tree, and Support Vector Machine are some of the algorithms employed in this research.

3.1.1 Logistic regression

It is employed as a classifier to categorize observations into several groups. Using the logistic sigmoid function, the approach converts its results to probabilities, which it then utilises to make a forecast for the goal. Unlike the simpler linear regression model, the Logistic Regression makes use of a sigmoid function rather than a logistic function. The goal of the logistic regression hypothesis is to keep the cost function between 0 and 1.

3.1.2 Algorithm of the random forest:

For each basic decision tree, a subset of all characteristics is only taken into account when splitting nodes, resulting in a model that uses three random concepts: randomly selecting training data when generating trees, selecting subsets of features when splitting nodes. As each tree learns from a random sampling of the data during training, a random forest is created. There are a lot of decision trees in a random forest model A forest is created by averaging the projected outcomes of individual trees. Three random notions are also incorporated into the algorithm: random selection of training data when generating trees, random selection of specific subsets of variables when dividing nodes, and only considering a percentage of all variables when splitting every basic decision tree. Every basic tree in a random forest learns from a random sampling of the dataset during training.

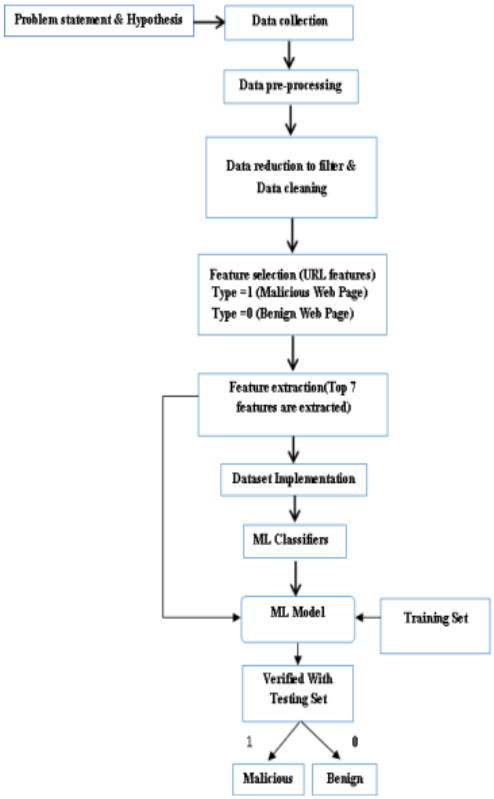
3.1.3 SVM Algorithm:

We employ a range of machine learning techniques, depending on the dataset, to predict and classify data. Linear models can be utilised to address classification and regression problems using the Support Vector Machine, or SVMs. It can handle both linear and nonlinear problems, making it helpful in a broad variety of contexts. Using a line or a hyper plane, SVM splits the data into several classes. Machine learning uses a kernel function known as the radial basis function kernel, or RBF kernel, in a variety of kernelized learning methods. Using support vector machines to classify data is a prominent application. It is possible to think of a hyper plane as an imaginary line that separates and sorts data in a classification issue using only two features as features (like the image above).

3.1.4 Tree of Decision

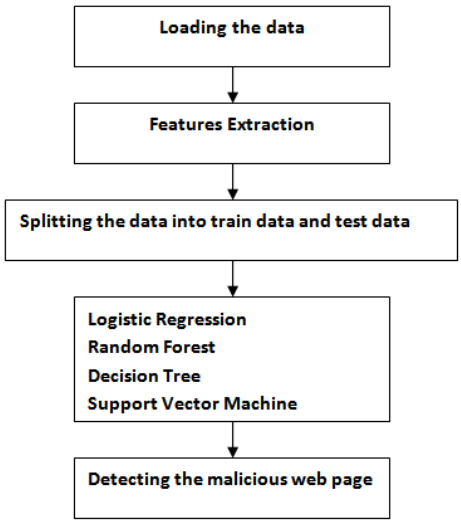
The goal is to create a model that can predict a target value by learning simple decision rules based on data attributes. This method has several advantages, such as being simple to perceive and understand or being able to solve issues with multiple outputs.

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 2, 2022, p. 2768 - 2775
https://publishoa.com
ISSN: 1309-3452



3.2 Framework/Architecture:

3.3 Algorithm and Process Design:



4 Goals and Their Execution

4.1 Collecting data

The quality of categorisation is strongly influenced by the datasets used. We need to pick a good dataset rapidly. By using the Kaggle database to collect both hazardous and benign websites, we fill this gap in our knowledge.

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 2, 2022, p. 2768 - 2775
https://publishoa.com
ISSN: 1309-3452

The F1 score, receiver accuracy, and the F1 score all serve as evaluation metrics Characteristics of the Operation Area We use the Receiver Operating Characteristics Area Under the Curve (ROC-AUC) metrics to evaluate the performance of our models. To calculate the F1-score and Accuracy, the FPR=False Positive Rate must be used to evaluate precision and recall.
True Positive Rate is referred to as TPR.
Accuracy
Precision
Recall
F1-score
The formula for calculating values is as follows: "True positive" (TP) means that the number of events has been correctly determined.
Number of events anticipated wrongly and not necessary (false negative, or FN).
FP = the number of occurrences that were incorrectly predicted.
If an occurrence has been correctly predicted but not necessary, it is known as a "true negative."
In machine learning, the False Positive Ratio (FPR) is a measure of how accurate the system is. In other words,

FPR = FP/(FP+TN) FPR/(FP+TN)

True Positive Rate (TPR): Because it is a synonym for recall,

TPR=FP/(FP+TN) is the formula used to compute it.

Accuracy:
In order to measure performance, it is important to know the ratio of successfully predicted observations to the total number of observations.
Recall: This is the ratio that properly anticipates positive observations in the original data from all observations.

Recall = TP/(TP+FN)

The values that have been appropriately detected are calculated with precision. To calculate this, take the total number of accurately predicted positive softwares and divide it by the total number of anticipated positive softwares. As previously stated, precision is calculated as

TP/(TP plus FP).

Model precision and recall are combined to provide the F1-score, which is defined as the average of the model's precision and recall. A score known as the F-score is another name for this.
The formula for the F1 score is 2(Precision Recall/Precision + Recall).

Another valuable measure for classifying problems is the ROC-AUC area under the prediction score ROC-AUC curve.

Implementation of the Code:

Feature Extraction
#loading only 100 samples (art websites data)

raw_data.head ()

	websites
0	http://www.emuck.com:3000/archive/egan.html
1	http://danoday.com/summit.shtml
2	http://groups.yahoo.com/group/voice_actor_appr...
3	http://voice-international.com/
4	http://www.livinglegendsltd.com/

#Here we divided the protocol from the entire URL. but need it to be divided it in to separate column

```
0          [http, www.emuck.com:3000/archive/egan.html]
1                       [http, danoday.com/summit.shtml]
2     [http, groups.yahoo.com/group/voice_actor_appr...
3                       [http, voice-international.com/]
4                     [http, www.livinglegendsltd.com/]
Name: websites, dtype: object
```

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 2, 2022, p. 2768 - 2775
https://publishoa.com
ISSN: 1309-3452

```
seperation_of_protocol = raw_data['websites'].str.split("://",expand = True) #expand argument in the split method will give y
ou a new column
```

	0	1
0	http	www.emuck.com:3000/archive/egan.html
1	http	danoday.com/summit.shtml
2	http	groups.yahoo.com/group/voice_actor_appreciatio...
3	http	voice-international.com/
4	http	www.livinglegendsltd.com/

```
#renaming columns of data frame
```

	domain_name	address
0	www.emuck.com:3000	archive/egan.html
1	danoday.com	summit.shtml
2	groups.yahoo.com	group/voice_actor_appreciation/links/events_an...
3	voice-international.com	
4	www.livinglegendsltd.com	

```
#Concatenation of data frames
```

	protocol	domain_name	address
0	http	www.emuck.com:3000	archive/egan.html
1	http	danoday.com	summit.shtml
2	http	groups.yahoo.com	group/voice_actor_appreciation/links/events_an...
3	http	voice-international.com	
4	http	www.livinglegendsltd.com	

We are importing the required package for the analysis and identifying the Hazard based on the datasets available from the site Kaggle and the dataset has been imported through pandas library and stored in the variable.
Then data has been extracted here we divided the protocol from the entire URL. But need it to be Divided it separate column

Features Extraction:

```
#Will show the results only the websites which are legitimate according to above condition as 0 is legitimate website
```

	protocol	domain_name	address	long_url
0	http	www.emuck.com:3000	archive/egan.html	0
1	http	danoday.com	summit.shtml	0
3	http	voice-international.com		0
4	http	www.livinglegendsltd.com		0
5	http	voicechasers.com	forum/viewforum.php?f=8	0

```
def have_at_symbol(l):
    """This function is used to check whether the URL contains @ symbol or not"""
    if "@" in l:
        return 1
    return 0
```

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 2, 2022, p. 2768 - 2775
https://publishoa.com
ISSN: 1309-3452

```
splitted_data['having_@_symbol'] = raw_data['websites'].apply(have_at_symbol)

splitted_data.head()
```

	protocol	domain_name	address	long_url	having_@_symbol
0	http	www.emuck.com:3000	archive/egan.html	0	0
1	http	danoday.com	summit.shtml	0	0
2	http	groups.yahoo.com	group/voice_actor_appreciation/links/events_an...	1	0
3	http	voice-international.com		0	0
4	http	www.livinglegendsltd.com		0	0

```
def redirection(l):
domain_registration_length_main('www.w3schools.com')

def age_of_domain_sub(domain):
    creation_date = domain.creation_date
    expiration_date = domain.expiration_date
    if ((expiration_date is None) or (creation_date is None)):
        return 1
    elif ((type(expiration_date) is list) or (type(creation_date) is list)):
        return 2
    else:
        ageofdomain = abs((expiration_date - creation_date).days)
```

The above-described function is used to categorise websites into three categories, and only legitimate websites will appear in the results.
If the URL has a symbol (//) after the protocol, it is considered phishing.

BeautifulSoup and whois are used to divide the dataset based on the domain and domain name and its length after feature extraction.

A csv file of the domain has been created for machine learning analysis.

DecisionTree_Classifier

```
import pandas as pd
legitimate_urls = pd.read_csv("extracted_csv_files/legitimate-urls.csv")
phishing_urls = pd.read_csv("extracted_csv_files/phishing-urls.csv")

legitimate_urls.head(10)
phishing_urls.head(10)
```

Data PreProcessing

```
urls = legitimate_urls.append(phishing_urls)
urls.head(5)
```

splitting the data into train data and test data
0.8287841191066998

Deploying ML algorithms for evaluating the accuracy value and applying samr procedure for other Ml
Support Vector Machine
0.7642679900744417

indices of columns : [10 5 12 6 4 8 2 7 11 3 9 1 0]

***Feature ranking: ***

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 2, 2022, p. 2768 - 2775
https://publishoa.com
ISSN: 1309-3452

Feature name :
Importance
1 statistical_report : 0.2702531638244638
2 URL_Length : 0.20712249310046107
3 web_traffic : 0.17522184222068443
4 age_domain : 0.08590997252246108
5 Sub_domains : 0.08138503997090331
6 domain_registration_length: 0.06574151936660255
7 Prefix_suffix_separation : 0.04458265918630407
8 dns_record : 0.03584104109668891
9 tiny_url : 0.02851732691424134
10 Redirection_//_symbol : 0.0028160428498986726

11 http_tokens : 0.0019997106549910324
12 Having_IP : 0.0006091882922996837
13 Having_@_symbol : 0.0

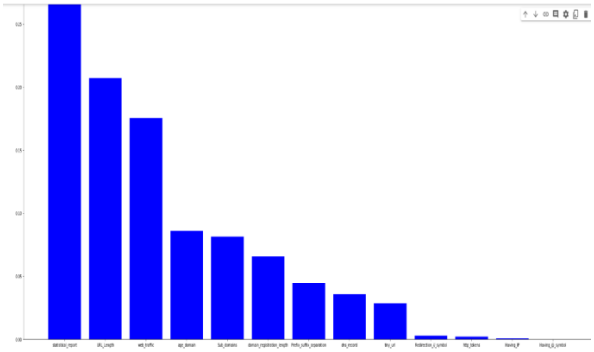


Fig- 1. Feature importance of the Random Forest classifier, along with their inter-trees variability

CONCLUSION

There has been a rise in the field of cybersecurity that focuses on identifying malicious web pages. Detecting fraudulent online pages has been the subject of numerous studies, but they are time and resource intensive, making them prohibitively expensive. In this study, we used machine learning techniques to classify web pages as harmful or benign based on features found in their URLs. Random Forest(RF) is a machine learning classifier that has a 95% accuracy rate. The results of our experiments suggest that our technology is capable of detecting malicious web pages.

REFERENCES

[1] Tao, Wang, Yu Shunzheng, and Xie Bailin. "A novel framework for learning to detect malicious web pages." In 2010 International Forum on Information Technology and Applications, vol. 2, pp. 353-357. Ieee, 2010.
[2] Eshete, Birhanu, Adolfo Villafiorita, and Komminist Weldemariam. "Malicious website detection: Effectiveness and efficiency issues." In 2011 First SysSec Workshop, pp. 123-126. IEEE, 2011..
[3] Aldwairi, Monther, and Rami Alsalman. "Malurls: A lightweight malicious website classification based on url features." Journal of Emerging Technologies in Web Intelligence 4, no. 2 (2012): 128-133..
[4] Yoo, Suyeon, Sehun Kim, Anil Choudhary, O. P. Roy, and T. Tuithung. "Two-phase malicious web page detection scheme using misuse and anomaly detection." International Journal of Reliable Information and Assurance 2, no. 1 (2014): 1-9.
[5] Hwang, Young Sup, Jin Baek Kwon, Jae Chan Moon, and Seong Je Cho. "Classifying malicious web pages by using an adaptive support vector machine." Journal of Information Processing Systems 9, no. 3 (2013): 395-404.
[6] Yue, Tao, Jianhua Sun, and Hao Chen. "Fine-grained mining and classification of malicious Web pages." In 2013 Fourth International Conference on Digital Manufacturing & Automation, pp. 616-619. IEEE, 2013..

[7] Krishnaveni, S., and K. Sathiyakumari. "SpiderNet: An interaction tool for predicting malicious web pages." In International Conference on Information Communication and Embedded Systems (ICICES2014), pp. 1-6. IEEE, 2014.
[8] Sun, Bo, Mitsuaki Akiyama, Takeshi Yagi, Mitsuhiro Hatada, and Tatsuya Mori. "Automating URL blacklist generation with similarity search approach." IEICE TRANSACTIONS on Information and Systems 99, no. 4 (2016): 873-882..
[9] Urcuqui, Christian, Andres Navarro, Jose Osorio, and Melisa García. "Machine Learning Classifiers to Detect Malicious Websites." In SSN, pp. 14-17. 2017.).
[10] Wang, Rong, Yan Zhu, Jiefan Tan, and Binbin Zhou. "Detection of malicious web pages based on hybrid analysis." Journal of Information Security and Applications 35 (2017): 68-74.74.
[11] Kim, Sungjin, Jinkook Kim, Seokwoo Nam, and Dohoon Kim. "WebMon: ML-and YARA-based malicious webpage detection." Computer Networks 137 (2018): 119-131.
[12] Altay, Betul, Tansel Dokeroglu, and Ahmet Cosar. "Context-sensitive and keyword density-based supervised machine learning techniques for malicious webpage detection." Soft Computing 23, no. 12 (2019): 4177-4191.
[13] website: http://jupyter.org/
[14] https://archieve.ics.uci.edu/ml/dataset/
[15] Ibrahim, M. Y. (2017). Real Time Xss Detection: A Machine Learning Approach.
[16]https://medium.com/thalus-ai/performance-metrics-forclassification-problems-in-machine-learning-part-ib085d432082b