# A HARMFUL WEB SITES DETECTION SYSTEM USING RANDOM FOREST MACHINE LEARNING ALGORITHM

***H.M.G.G. Dammika*, D.D.M. Ranasinghe***

*Department of Electrical and Computer Engineering, The Open University of Sri Lanka*

## INTRODUCTION

During this modern era, people around the world use the internet extensively for work and other purposes, and the usage exponentially increased during the pandemic. Hence, there is a rapid upgrading of the entire related infrastructure, including connectivity with increased attention to web security. It is reported that people have come under potential attacks from malware, viruses, and other harmful agents (Jang-Jaccard & Nepal, 2014). Thus, identifying and dealing with infected websites, before a normal user accesses them, becomes a top priority for the safety of legitimate users. According to the Google safe browsing report, on 8th May 2022, about 4,819,522 harmful websites are being blacklisted weekly by Google search (Google Transparency report, May 2022).

The most common methods to identify harmful websites are the Blacklisting approach and the Rule-based Approach (Lekshmi & Thomas, 2019). Current approaches to dealing with this problem have many limitations in terms of effectiveness and efficiency. The blacklist method is having a database consisting of a set of web addresses that were harmful in the past (Lekshmi & Thomas, 2019). Since blacklisting is time specific, it cannot report an exhaustive list and lacks the ability to detect newly generated harmful web addresses after a specific time. Maintaining a list of harmful website addresses is also very difficult because hackers modify the original website addresses with minute changes to avoid being blacklisted.

The rule-based Approach uses two sub-methods to identify harmful web addresses (Lekshmi & Thomas, 2019). The two sub-methods are Signature based method and Behaviour based method. In the Signature-based method, the defined signatures are protocol TCP, service HTTP, and context host. This is an extension to blacklist methods, where the idea is to create signatures for harmful websites. Then the signatures are matched and tested to find the relationship between the new web address and the signature of an existing harmful web address (FortiGate, 2007). Limitations in these methods are requiring to a high amount of manpower, time, and money to extract unique signatures.

The behaviour-based detection method is based on the behaviour property of executable files. This behaviour-based detection method has a data collector, an interpreter, and a matcher. The interpreter is used for translating data collection module into intermediate representation. The matcher examines the above representation with behaviour signature (Pingle et. al., 2020). Limitations of this approach are high scanning time and storage complexity for behavioural patterns.

In addition, there are different machine learning techniques which are used to classify harmful websites using features and behaviours taken from website addresses, web content and network activity. The detection methods and tools which adopt this approach of patrolling web content may consume more computation time and resources. The other main concern about it is the difficulty in collecting features and behaviours. The choice of the appropriate set of features is very important for the quality of the classifier's performance. But it was found that the previous studies mainly applied features such as the host-based and page content-based features (Towards Data Science, 2021). Content-based features are statistical

information extracted from the raw content such as HTML and JavaScript content of the web page, and the statistical information is number of scripts, length of HTML and page entropy. Host-based features are obtained from the host-name properties of the web address. Some of these properties are registration date, registration country, life remaining etc. Harmful websites usually hide information pertaining to registration and they usually use the same host IP to create many harmful websites at the same moment. Collecting some of these features is time-consuming, and some feature data could be inaccurate due to various types of tampering. Hence, it is necessary to find an easy and efficient method of collecting the required features for the analysis.

The paper (Sahoo et. al., 2017), shows that the lexical structure of harmful web address strings is observed to be significantly different from that of benign web address strings. For example, harmful web addresses on average have several levels of sub-domains, special characters in the web address path. Usually, their domain names resemble whitelist domains, but with minor changes, higher average domain lengths and patterns which are not usually seen in benign web addresses. Hence, lexical features have the potential to distinguish between harmful and benign web addresses and can be used to select features for classification.

The use of lexical features with the classification model will yield high system accuracy and will provide an easy and efficient method for feature collection in web addresses. The main advantage of the proposed system is that it can predict harmful websites as well as identify the type of harmfulness associated with the website.

Hence, this research focuses on building a model to detect harmful websites using machine learning techniques based on the lexically based attributes of the website. This will help in safe web usage and a better user experience. By properly reporting harmful websites, users can avoid significant privacy violations. Users will also be able to prevent any illegal activities that may inadvertently involve them. Identifying harmful websites will also help users to avoid being sufferers of attacks that use blackmailing and false information to get the financial advantage of their loser.

## METHODOLOGY

The overall research design to address the problem adopted two layered system architecture is given in figure 1. In the first layer, it was proposed to identify whether a particular website is harmful or not. In the second layer, the type of harmfulness of the website was detected.

The input data set for training the model comprised around 120,000 web addresses, which were collected from various standard websites that provide categorized web addresses. Some of these websites are phishtank.com, openphish.com, joewein.de.htm,unb.ca, majestic.com. The data set consists of benign as well as harmful web addresses in four categories namely Spam, Malware, Phishing and Defacement. During the pre-processing stage, unnecessary columns were removed and what was retained were columns containing details of web addresses, statutes of harmfulness, and the type of the harmfulness. In the second step feature extraction module for HWDS was created. Since a web address consists of a string of characters that cannot be directly fed into a machine learning algorithm it is essential to extract suitable features for further processing. The extracted features are Lexical features, which are used for mathematical interpretation of the model. Lexical features are the static properties of the web address such as URL length, path length, hostname length etc. These features were collected by the urlparse library in python which provides a quick way to break down web addresses into their individual components.

The training data set consisted of 80% of the total data set and the rest of the 20% was used for testing purposes. The model training was done using an ensemble machine learning

approach and trained the data set with Random Forest, Decision Tree and K Nearest Neighbour algorithms.

Features of harmful and harmless website addresses are having different distribution patterns hence it is vital to provide these features correctly for accurate model building. The features that are obtained by feature extraction are further processed into a numerical format using the term frequency-inverse document frequency (TF-IDF) vectorizer and then fed for the training model. After creating the model the type of harmfulness in the website is identified based on the numerical format feature extracted data.

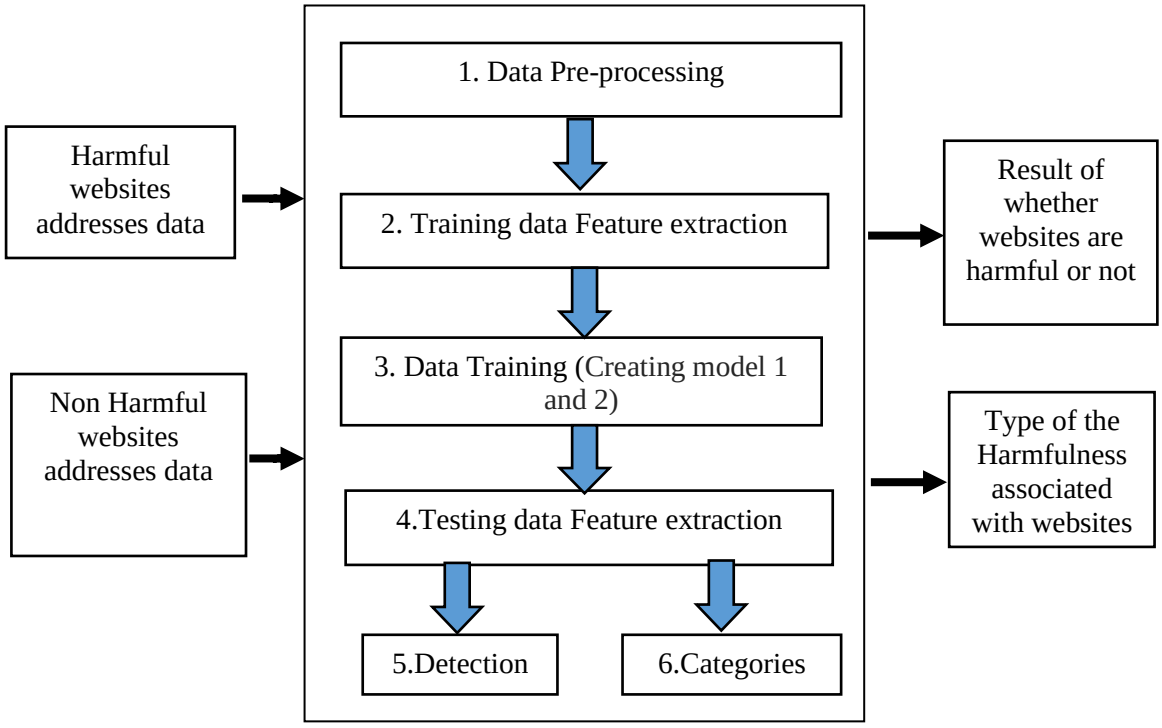


Fig 1: Overall system design

In the fourth step, a user can input a web address into the system. The developed system will be able to extract features from the user's input web address and classify whether the web address is harmful or not. In the case of a harmful web address, the type of harmfulness will be detected.

**RESULTS**

The research demonstrates a two-layered approach to detecting harmful websites and the classification of harmful websites. In layer one, a website is detected for harmfulness and is a binary classification problem with two classes, namely benign and harmful. In layer 2, the type of harmfulness of the website is identified, and it is a multi-class classification problem where harmful webpages are categorized into specific classes such as: defacement, malware, phishing, and spam.

In training the model Decision Tree algorithm produced 98.1% accuracy, the Random Forest algorithm produced 99.8% and the K Neighbors algorithm produced 94.5% accuracy. Considering the high accuracy rate produced by the Random Forest algorithm, the precision

and recall were calculated based on the F1 score as shown in figure 2 below. Both the precision and the recall of the model are 0.998.

```
Model accuracy =  99.78412970589586
[[24539    43]
 [   98 40637]]
              precision    recall  f1-score   support

        good       0.996     0.998     0.997     24582
     harmful       0.999     0.998     0.998     40735

    accuracy                          0.998     65317
   macro avg       0.997     0.998     0.998     65317
weighted avg       0.998     0.998     0.998     65317
```

Fig 2: Performance of the Random Forest model

The simulation result of feature extraction for system design is shown below.

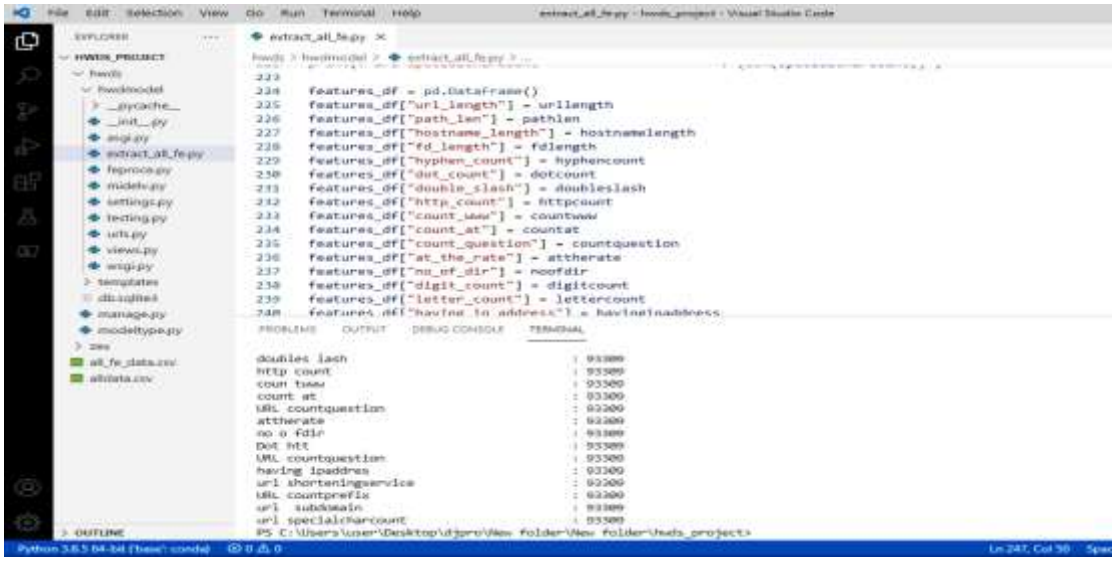


Fig 3: Feature extraction for HWDS model

Fig 3 shows, the collected web address data (for model 1 training) and are sent to the feature extraction python file and then generates a list of feature data as the output.

## DISCUSSION AND CONCLUSION

This research explored random forest machine learning techniques based on the lexical features approach to identify whether a particular website is benign or harmful. If harmful, the type of harmfulness is also identified**.** This technique is an add-on to the former method of blacklisting. Selected feature sets applied on random forest classification yield a classification accuracy of 99.8% with a low false positive rate. In addition, it can be extended to calculate the risk rating of a malicious web address after parameter tuning and learning with huge training data. The users are able to interact with the system through the web interface where the user can input the suspicious web address for classification. This will help in safe web usage and a better user experience. The end result of this project will also help users to avoid being victims of attacks that use blackmail and false information to get the financial advantage of their victims.

**REFERENCES**

FortiGate, (2007). Retrieved from https://community.fortinet.com/t5/FortiGate/Blocking-URLs-with-IPS-signatures/ta-p/19171

Google Transparency report, accessed on 08.05.2022. Retrieved from https://transparencyreport .google.com/ safe-browsing/overview

Jang-Jaccard J., Nepal S., (2014). A Survey of emerging threats in Cyber Security. Computer and System Sciences, 80 (2014) 973–993.

Lakshmi R.A., Thomas S., (2019). 'Detecting malicious URLs using machine learning techniques: a comparative literature review', International Research Journal of Engineering and Technology-ISSN: 2395-0072.

Pingle Y., Patil S., Bhathkar S.N., Patil S., (2020). Detection of Malicious Content using AI. Proceedings of the 14th INDIACom; INDIACom-2020; IEEE Conference ID: 49435

Sahoo D., Liu C., Hoi S.,(2019). 'Malicious URL Detection using Machine Learning: A Survey', School of Information Systems, Singapore Management University, Vol. 1, No. 1, Article

Towards Data Science, (2021). Retrieved from https://towardsdatascience.com/extracting-feature-vectors-from-url-strings-for-malicious-url-detection-cbafc24737a