Wireless Personal Communications (2022) 125:3227–3248
https://doi.org/10.1007/s11277-022-09707-w

APuML: An Efficient Approach to Detect Mobile Phishing Webpages using Machine Learning

Ankit Kumar Jain[1] · **Ninmoy Debnath**[1] · **Arvind Kumar Jain**[2]

Accepted: 10 April 2022 / Published online: 2 May 2022

Abstract
Nowadays, the growth of mobile phones users has gained a significant increase because of the features offered by them in abundant amounts. These devices are being used rapidly for accessing the web and many online services. However, the security mechanisms that are available in smartphones are not yet mature. Therefore, smartphones are vulnerable to various types of attacks, such as phishing. The browsers on smartphones are very trivial and the smartphones security abilities have been lessened, to match the smartphone's capabilities. Therefore, detection of the malicious website is different from the previously known technique, which is used on the desktop. Many anti-phishing techniques for mobile devices have been developed but still, there is a lack of a full-fledged solution. Therefore, this paper presents an efficient approach to detect malicious mobile webpages. The proposed approach APuML (Anti Phishing using Machine Learning) extracts all the static and site popularity features from the given URL to create a feature vector. An appropriate machine learning classification algorithm is then applied on the feature set to obtain the result and update the database accordingly. In our approach, the Random Forest classifier outperforms over other classifiers and achieved detection accuracy of 93.85%. We have also created an endpoint application for the users to interact with our system using his/her mobile devices. Moreover, the proposed approach can identify drive-by downloads attack, zero-day attack and clickjacking attack with high accuracy.

Keywords Smartphones · Malicious mobile webpages · Machine learning · Phishing

✉ Ankit Kumar Jain
ankitjain@nitkkr.ac.in

Ninmoy Debnath
ninmoy1994@gmail.com

Arvind Kumar Jain
arvindjmp@gmail.com

[1] National Institute of Technology Kurukshetra, Kurukshetra, India

[2] National Institute of Technology Agartala, Agartala, India

1 Introduction

1.1 Context

In today's world, the Internet has made a drastic change in our lives, providing almost everything with just one click of a button. The existence of many online services such as e-marketing, e-banking, e-shopping, contactless payments, etc. have made our lives more convenient by permitting these services to manage our transactions while sitting at home. In this way, mobile devices are being used rapidly to access these online services i.e., accessing the web because of the enormous spread of mobile phones. Figure 1 shows all global website traffic was generated by mobile phones from 2011 to 2020 [1]. As stated by a report [2] released in August 2019, 1.56 billion mobile phones were sold worldwide. They are mainly used because of their portability and their capability to render the facilities that personal computers render. Due to this, mobile devices are being exposed to many security threats. However, despite considerable advances in bandwidth and processor, the browsing experience on smartphones is significantly different. These dissimilarities can mainly be appertaining to the substantial depletion of the size of the screen, which influences the functionality, layout, and content of mobile webpages.

Since the contribution of mobile phones in the generation of web traffic is getting increased, therefore all search engines have also become progressively mobile-focused. Since search engines have shifted to mobile users behaviour, so do the websites and advertisers.

1.2 Problem Definition

Phishing is the most dominant attack on mobile phones among all of the attacks which are used to lure the victims with the objective of gaining their private information. Wombat Security's State of phish 2018 report indicated that at least three-quarters of organizations were targeted by phishing URLs attack through their mobile phones [3]. In 2018, a report from Internet Crime from the Internet Crime Complaint Center (IC3) showed that $48,241,748 was lost per victim due to phishing URL attacks in the same year [3]. In 2019, an attack on mobile phones started with a simple notification sent to the targeted victim's email address saying that 'your email account is accessed by a new device'. There was a button in the notification prompting to review the activity, which led the victim to a phishing page instead. When this page was visited on a mobile phone, the page seemed to be

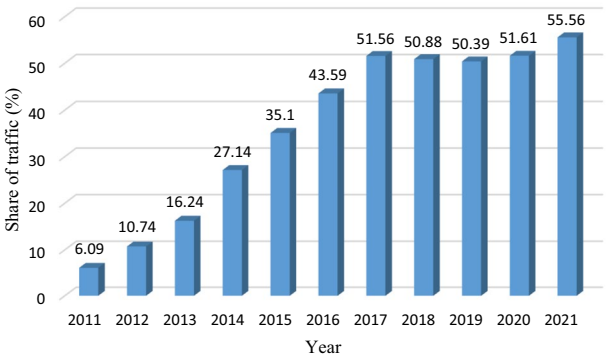

Fig. 1 Global mobile phone website traffic share

benign and that led to the Google landing page. However, when visited on a desktop PC, it became clear that the page is illegitimate because of the fact that the link address bar became filled with a lot of random text and to hide the actual link in mobile phone Google translate was used to serve the landing page [4].

In the current situation in the world, amongst the fears of the COVID-19 pandemic, malicious websites are speeding up [5]. Attackers have created a Coronavirus tracker map to spread malware that focused on stealing information from users. A phishing website was created, 'corona-virus-map.com' that seemed to be a benign live tracking map for the COVID-19 virus. After identifying this attack, HC3 (Health Sector Cybersecurity Coordination Center) reported that the attackers were imitating Johns Hopkins University, a reputed health institution, to contaminate visitors with the AZORult trojan. The mobile phones having limited security in the browser were infected in a greater amount [6].

The mobile-specific websites are contemplated as a classic domain for the various types of attacks, for the succeeding reasons:

- The web pages can easily be forged by just duplicating the source code of the website for falsification.
- Many complex anti-phishing mechanisms that can work accurately for desktop cannot work accurately for smartphones or mobile devices due to the limited resources of smartphone devices [7].
- The size of the screen of our smartphones is small compared to the desktop; therefore, it has become very much problematic to detect malicious websites only because of their appearance or through any security indicators.
- The smartphone browsers are very trivial, and the smartphone's security abilities have been lessened for suiting the smartphone abilities [8].

There are many existing techniques to detect and prevent these malicious attacks. However, existing techniques do not work accurately for mobile devices because mobile devices have limited resources. Therefore, there is a requirement for real-time, fast, and intelligent solution for this problem [8].

The domain names of the fake webpage could be easily rendered identical to legitimate websites. Figure 2 shows legitimate and fake webpages of "PayPal". Attacker changed "l"(small L) with "I" (Capital I) and it is difficult to distinguish because these two letters look identical as shown in the figure. By using this kind of letter substitution attacker often build malicious URL.

1.3 Proposed Solution

To solve the mobile phishing webpage detection problem, this paper presents a machine learning based approach to detect malicious sites on mobile phones. It aims to build an android application and design a mechanism, which is fast to determine malicious mobile webpages. This mechanism makes use of site popularity and static features of mobile webpages, extracted from their HTML, JavaScript content, Uniform Resource Locator (URL) and advanced smartphone abilities. In the next phase, the approach creates a feature vector. It then calculates the benign and malicious probability of the URL using Naïve Bayes and adds them as a feature. An appropriate machine learning algorithm is applied to the feature vector and then the results are obtained. After that, the given URL is added to the database with its type (malicious or non-malicious). We have collected 4000 mobile benign and

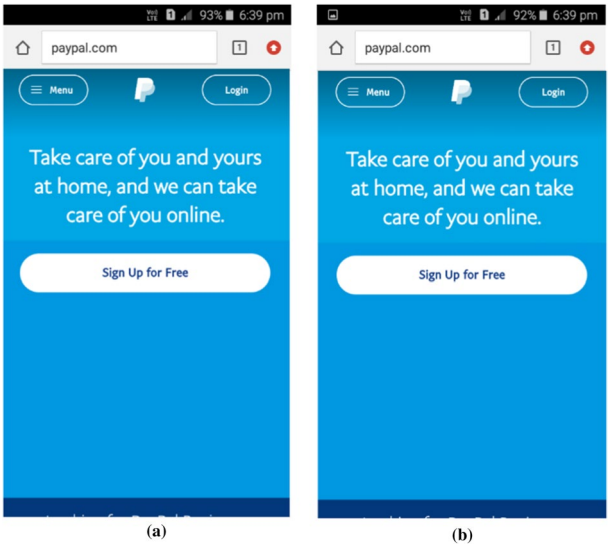

Fig. 2 **a** original PayPal mobile webpage and 3 **b** fake PayPal mobile webpage

malicious webpages from various sources such as openphish.org, phishtank.org, and alexa.com. We then use different classification technique such as Logistic Regression, K-Nearest Neighbors, Support Vector Machine, Random Forest, Decision Tree, Neural Network to develop a model. Finally, we choose the random forest model which provides 93.85% accuracy and 93.22% true positive rate. Our model also detects a range of malicious mobile webpages that current techniques such as Google Secure Browsing do not precisely detect. We have also compared our approach with existing mobile phishing detection approaches on various parameters.

1.4 Contributions

The major contributions to the proposed method are as follows:

- The proposed approach identified efficient features in the detection of mobile phishing webpages.
- The proposed approach can identify drive by downloads attack, zero-day attack and clickjacking attack with high accuracy.
- Proposed approach can detect malicious mobile-specific websites on mobile phones written in any textual language.
- We have also conducted a feature analysis process to predict the most powerful features in the detection of malicious websites on mobile phones.

1.5 Outlines

The remainder of this paper is organized as follows. Section 2 presents the related work. Section 3 describes our proposed approach in detail and presents the extractions of various features to train the machine learning algorithms. Section 4 presents the data collection and

implementation details. Section 5 presents features analysis and results outcomes. Finally, Sect. 6 concludes the paper and presents future work.

2 Related Work

Various type of web-phishing detection techniques have developed. Website phishing detection techniques are classified into different categories such as machine learning based phishing detection, HTML based phishing detection, OCR based phishing detection, login form-based phishing detection, search engine base phishing detection technique, etc. This section discusses phishing detection in the mobile domain and desktop domain.

2.1 Mobile Phishing Detection Approaches

Amrutkar et al. [7] proposed a technique "kayo" which uses a total of 44 features. These 44 features are categorized into four types namely mobile-specific, JavaScript based, HTML features, and URL specific features. The author also showed that every feature is significant and influences the classification result. The authors used the bolster vector machines learning algorithm for classification. The authors have taken a data set of 349,150 non malicious URL and 5,231 malicious URL's and get a 90% detection accuracy, 89% true positive rate, and 8% false-positive rate.

Rao et al. [9] developed an android application called PhishDump to classify the legitimate and phishing webpages on mobile devices. PhishDump is based on the multi-model ensemble of Long Short-Term Memory (LSTM) and Support Vector Machine (SVM) classifier. The techniques used URL based features, and achieved detection accuracy of 97.30% on their data set and 98.50% on the benchmark data set.

Bottazzi et al. [10] proposed an approach MP-Shield which used two techniques, public blacklist search and machine learning for the detection purpose. The MP-Shield contains high-level architecture that is made of three objects. (1) blacklist API, (2) machine learning classification engine, and (3) watchdog. The first object search query in Google Safe Browsing service. Second, run a set of machines learning model based on the WEKA software. Watchdog does these two works in parallel.

MP-Shield (mobile application) inspects network traffic for phishing detection using the VPN service, The VPN service analyzes IP packets before relaying them to the original target hosts. For each unclassified GET request, the proxy passes it to the Watchdog engine, which runs asynchronously concerning the browsing process to reduce latency. The copied packets are buffered and analyzed by the Watchdog, and target URLs are extracted from the obtained HTTP requests to be classified. Approach achieved detection accuracy of 89.2% and TPR 89.2% using J48 algorithm.

Johny et al. [11] proposed a content matching approached for the detection of the malicious webpage. In this, the screenshot of a loaded webpage is matched with the database of legitimate pages. The approach uses the service of AWS which runs the scale-invariant feature transform (SIFT) algorithm for matching webpages. When a new URL that is not presented in PhishTank [12] is loaded in the user's phone, the screenshot of the loaded website is taken and compared with SIFT algorithm. If the webpage is matched then the loaded image is marked as safe else, the webpage classified as phishing. Approach achieved detection accuracy of 92%. However, it is not possible to store all the legitimate websites.

2.2 Phishing Detection in Desktop Domain

There are many proposed solutions for detecting phishing websites in the desktop environment. The broad classification of those schemes based on underlying techniques are mainly list based methods like blacklisting or whitelisting, or visually comparing various sites, based on DNS, proactive phishing URL detection based and machine learning based methods.

Whitelisting or blacklisting [13] scheme consists of either a list of legitimate websites known as whitelist or a list of aberrant sites known as blacklist in order to recognize phishing sites. The basis of classification is the response provided by various users or various third parties who perform detection in various ways. This is a real-time and a lightweight solution which can be implemented in browser but issues were the repeated improvisations of either of the list which cost very exorbitant.

Visual Similarity [14] based technique examines for visual alikeness among various webpages for checking phishing. First given website is matched with the present set of all websites based on their visual features. Then it is examined that if the given URL is present in the genuine set or list of websites if it is found in this authentic list then URL is considered as legitimate else declared as legitimate. This approach needs high computational resources and also high storage for the matching of visual contents. The time to load sites on the browser also increases gradually as the actions performed during detection are quite time-consuming.

DNS based approach [15] constitute validating the IP address of a suspicious website as the requested URL will be sent to DNS which verifies whether this IP address is present in a set or list of IPs of genuine sites. If the IP is not present in the list, the site is considered a malicious one. Problems with this technique include exorbitant communication cost, also this method is unable to work properly in case of DNS poisoning. This method also increases the burden on DNS.

Machine learning techniques [16] train the classification algorithm using various features that can effectively differentiate a legitimate instance from a phishing instance. The performance of these techniques depends on the features set, classification algorithm and training dataset.

3 Proposed Approach

The objective of this work is to create an application that checks the legitimacy of the URL entered by the user. Figure 3 presents the system architecture of the proposed approach. The proposed approach is mainly divided into three stages. The first stage checks whether it has already been verified before. In the second stage, various features have been extracted from the URL and the last stage classifies whether the website is legitimate or a malicious one and update in the database accordingly.

Stage 1: The system maintains its own database, which contains all the URL of the websites that has been already checked by the system and reported under the malicious or legitimate category. Whenever a user clicks for a website, the approach first determines through the database if the particular URL has already been checked. If yes, it informs the user about the legitimacy of the website accordingly. If the URL does not exist in

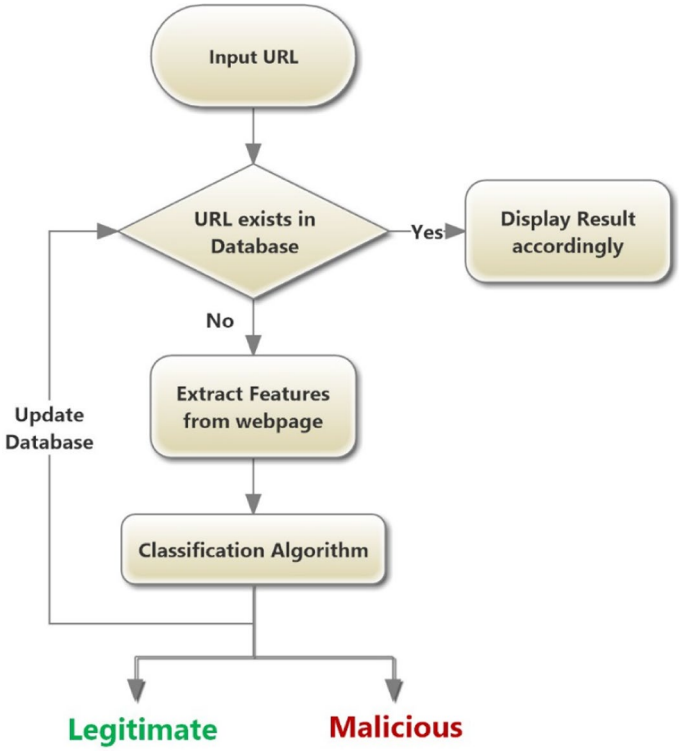

Fig. 3 System Architecture of Proposed Approach

the database, then the system enters stage 3 for further analysis. This process minimizes the wastage of time before starting the analysis.

Stage 2: In this stage, the application sends the URL to a server. It extracts a set of features that will be used to decide the legitimacy of the website. There are many feature groups, which have been considered by the system to form the feature set as described in Sect. 3.1. These groups constitute related features that have been taken by studying various proposed solutions and their effectiveness. Feature analysis is performed to get important indicators for the model and then various classification algorithms have been applied and evaluated which will classify the URL into either a legitimate or a malicious site. After the detection, the system proceeds to the final stage.

Stage 3: In this stage, the detected website URL will be added to the database handled by the server for future use so that if the same URL is being requested by the user again, it can be detected at an earlier stage. Now, the result is sent to the application which displays it to the user.

3.1 Feature Extraction

A webpage has many components, which includes HTML, JavaScript, images, and unified resource locator (URL). The webpages running on mobile devices can further communicate with different apks in the user's device using different tools. Our focus is on extracting

mobile relevant features from the URLs collected in the data collection process because we believe that they are notable indicators for the legitimacy of the website. We have considered 34 features as illustrated in Table 1. It is important to note that the value (magnitude) of a particular feature of the mobile specific webpage is different from that of the desktop version of websites. Example: The number of redirections in the mobile-specific webpage is higher than that of the desktop version, Number of JavaScript present in the desktop version of the webpage is a little bit higher than that of the mobile version of the webpage.

- *Mobile Specific features*: Three features that are specific to the mobile devices are considered consisting of- Number of SMS API calls, Number of tel API calls, and Number of apk API calls. These are considered so that the potential of the device's webpages can be recorded. These webpages have access to the user's personal data. They communicate with the applications on the user's device using web APIs like tel: and sms: which allows them to liaise with the phone and sms built-in applications, respectively. There are popular markets like Play Store where anyone can launch any of their application and if a webpage publishes their own application(.apk), this suggests bad behaviour. This kind of behaviour is also captured.
- *JavaScript Features*: JavaScript (JS) allows user to enhance webpages by making it more interactive. It consists of many built-in functions, which adds functionalities to the static webpages. Seven features are recorded so that we can apprehend all the JS related behaviour. We determine whether any JavaScript is embedded into HTML between the pair of script tag. We also determine the snippets of JS present in the main script or imported into them. We also found that the presence of "nonScript" is higher on a genuine webpage, wherein presence "scripts" is higher in the malicious webpages. As mobile is a light-weight device and recent times have shown that the highest number of attacks on them are phishing, we believe there are higher chances that a legitimate webpage may contain snippets that are written outside the main script and then imported into them. We also capture the presence and number of no-script tag. By analysis, we got to know that non-malicious websites contain more external JavaScript as compare to malicious websites.
- *HTML features*: Hypertext markup language (HTML) is the most basic language required to create a webpage. It is used for making static webpages. Seven relevant features from it have been considered to be extracted. Popular webpages use images, embed different snippets, etc. to create a more interactive webpage. Accordingly, we first check if the page has any images or links and then determine the number of them. We have researched that attackers insert links to malicious code in iframe [7] which contribute highly in case of drive-by downloads and clickjacking attacks. It has been noticed that the iframes are scattered in different patterns when compared between mobile and desktop webpages. It also has detected that webpages that take several redirections have a higher chance to be malicious because this prevents them from being identified by their DNS. All these information have been considered to be recorded.
- *URL Features*: We have observed that the Unified Resource Locators are different for benign and non-benign webpages. The mobile device cannot display the whole URL to the client. This is a vulnerability which has been exploited by the attackers. Therefore, nine URL related features (like length, number of digits, etc.) have been extracted so that all the differences can be recorded.
- *Website Specific Features*: This represents the traffic rank of the site all over the world and in the country in which the program is run i.e. in our case, the traffic rank of India is added as a feature. These two features are acquired from Alexa.

Table 1 Details of features of approach

Category	Features	Total Number of features
Mobile specific	Number of SMS API call, Number of tel API call, Number of apk API call	3
JavaScript	Presence of JS, Presence of NS, Presence of external JS, Number of JS, Number of NS, Number of external JS, Number of internal JS	7
HTML	Presence of image, Presence of iframe, Presence of links, Number of image, Number of iframes, Number of redirects, Number of links	7
URL	Length of URL, Number of forward slash, Number of question marks, Number of dots, Number of hyphens, Number of underscore, Number of equal signs, Number of ampersand, Number of digits	9
Website specific	Presence of World Traffic rank, World Traffic rank, Presence of Country Traffic rank, Country Traffic rank	4
Fake form	Presence of fake form, Number of fake form	2
Naive Bayes	Naive Bayes probability0, Naive Bayes probability1	2
Total		34

com. Intuitively, malicious webpages are less popular than benign webpages and so, this feature acts as a strong indicator. Most of the malicious websites are live for a maximum of 2 to 3 days. Therefore, gaining a good rank in alexa.com is a little bit impossible. Therefore, these two features work well for malicious website. However, sometimes these features may give a false-positive result for a newly created good website. Algorithm 1 is used to find the website specific features.

Algorithm 1: Finding website specific features
Step 1: find the rank of URL from alexa.com (country specific and world specific rank) *Step 2:* if country specific rank is present then *Presence of Country Traffic rank* = True *Country Traffic rank* =#country specific rank else *Presence of Country Traffic ran k*= False *Country Traffic rank* = 0 *Step 3:* if world specific rank is present then *Presence of World Traffic rank* = True World Traffic rank =#world specific rank else Presence of World Traffic rank = False World Traffic rank = '0

- *Naive Bayes Probability*: The probability of a URL being benign or malicious is calculated using Naive Bayes classifier and added as a feature. It is a collection of classification algorithms based on Bayes theorem. It is a simple and fast algorithm and works well even when there is a small amount of training data. It gives an advantage of using two algorithms at the same time, Naïve Bayes being the supporting algorithm. Probability0 is means webpage is legitimate and probability1 for the malicious webpage. Algorithm 2 is used for setting naïve based probability.

$$P(y|x_1, \ldots x_n) = \frac{P(x_1|y)P(x_2|y) \ldots P(x_n|y)P(y)}{P(x_1)P(x_2) \ldots P(x_n)} \tag{1}$$

where × 1…….xn are feature set.

Algorithm 2: Setting Naive Bayes probability feature
if Naive Bayes probability0 >= 0.5 then Naive Bayes probability0=True Naive Bayes probability1=False else Naive Bayes probability0=False Naive Bayes probability1=True

- *Fake Form*: we have collected two features from related to login form. Attacker create fake login form or any other form for taking information from the users because it is the only way to obtain the user's personal data. If there is no any form in the web-page then the web-page is safe.

4 Dataset Collection and Implementation Details

4.1 Data Collection

The data collection process includes the gathering of labelled benign and malicious mobile-specific webpages. We first, understand and identify what are mobile-specific webpages. Mobile-specific webpages include the webpages which have different URL in mobile and desktop browser. We analyze the URLs of such pages and identify the significant characteristics of them as explained in Table 2. We have manually collected 2000 benign URLs by obtaining popular websites from Alexa [17] from an Android mobile browser. For the malicious URLs, we have collected 2000 mobile-specific malicious URLs from OpenPhish [18] and phishtank [12] using an algorithm based on the characteristics shown in Table 2. This results in a dataset of size 4000.

4.2 Implementation Details

We have implemented our model using python. Different python libraries are used for running the code are as follows:

a) *BeautufulSoup*: Beautiful Soup used for parsing HTML and XML documents.
b) re: re library is used to fit one string with another string for a regular expression.
c) *Sklearn:* Scikit-learn (Sklearn) is the most useful and easy to used library for machine learning in Python. It provides many predefined functions that can be easily used for machine learning. It includes different model for regression, classification, dimensionality reduction, and clustering.
d) *Pickle:* Pickle library in python is used for serializing and de-serializing a Python object structure. Pickel library is used to save any object to the disk so that object can be used later on simply importing from that pickle file.
e) *Requests:* Requests is a Python library that is used for making kinds of HTTP requests. This library is extremely easy to use as it contains many predefined functions that are used for passing parameters in URL, SSL verification.

Table 2 Mobile Webpage Identifiers

Top Level Domain	.mobi
Sub domain	m.,mobile.,touch., 3 g.,sp.,s.,mini., mobileweb.,t
URL Path Prefix	/mobile, / mobileweb, /m, / mobi, /?m = 1, / mobil, /m_home

f) *Urllib:* Urllib library has different URL handling function for python. such as urllib.request for opening the URL and reading the URL.urllib.parse is used for parsing URLs.
g) *Flask:* Flask is an easy to used python library that consists of the web framework, Flask library is used to create web applications.

4.3 Flask API

We have created an API (endpoint) for the saved model using Flask micro web framework. An android application acts as a client in which the user can enter the URL. The android application is implemented in android studio and an emulator having Android 6.0 with API 23 has been used to test and run the application. It returns a string stating malicious or non-malicious as a response. A database containing already verified URLs is stored in a file. When the server starts, it reads the contents of the file and converts it into a dictionary. The URL when received from a client in the form of JSON object, is first checked through the dictionary and a faster response is given when the URL is found in it. If the URL doesn't exist in it, the legitimacy of URL is checked through a machine learning model. Then, URL is updated in the dictionary, and file so that the result of it can be retained throughout the server session and even after the server restarts, respectively. The connection of API with users and database are shown in Fig. 4.

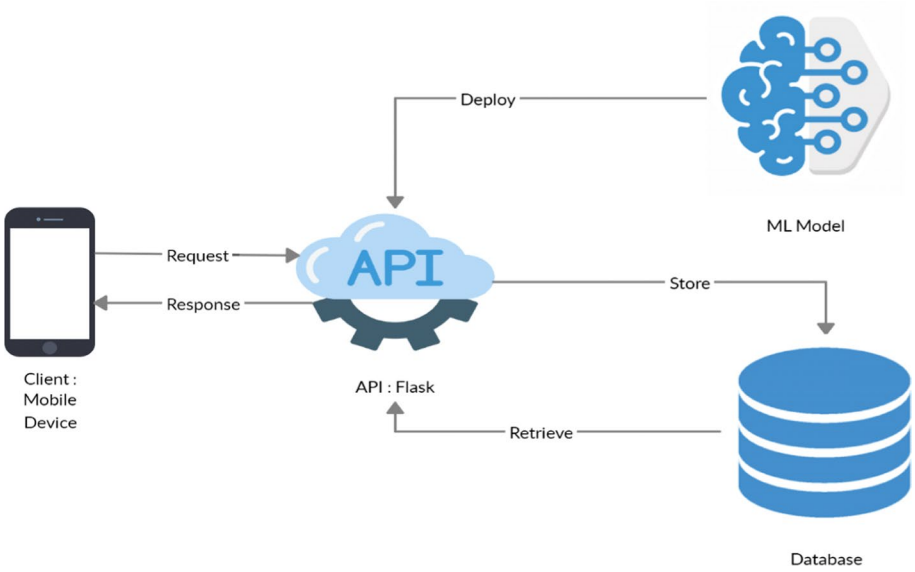

Fig. 4 Flask API Implementation

5 Features Analysis and Results Outcomes

5.1 Feature Selection

Our model used a total of 34 features as discussed in Sect. 3. We have divided these features into two subcategories as shown in Table 3. First is binary category features which contain nine features. Another category of numeric features gives value in numeric form. These 23 features are used to calculate Naïve Bayes probability features. The 23 that have selected for calculating naïve base probability are quantity base features (e.g., world traffic rank feature gives value 99,999 for same website www.test.com) and feature value may be exceptionally large as compare to the value produced by binary category feature. Therefore, the features that produce numerical values may be dominated over other features and gives false positive results. We have used Naïve Bayes classifier for these feature set as this classifier assume that the value of a particular feature is independent of the value of any other feature for given the class variable [19]. When these 23 features placed in the trained Naïve Bayes model, the model gives two outputs, Probability0 and Probability1. Probability0 implies the probability of non-maliciousness and Probability1 implies the probability of maliciousness of the website. If the probability of any website is more than 50%, we replace that percentage with value 2 otherwise; our model replaces it with value 0. We replace it with 2 because from the mutual information graph, we know that the dependency of output to this feature is very high. Our final features set consists of 11 features including 9 binary features and 2 Naïve Bayes probability features as shown in Fig. 5. After extraction of these features, a feature vector is created corresponding to each website for finding legitimate and phishing webpage.

5.2 Feature Analysis

We have done a feature analysis process using a feature importance method. We have extracted all the relevant features as discussed in Sect. 3.1. Then, we have analyzed all the features and decided the most prominent features that give the highest accuracy by trial and error using Mutual Information (MI) algorithm [20]. Mutual information measures the dependency of one variable to another (i.e., input feature and output) by quantifying the amount of information obtained about one feature, through the other

Table 3 Division of Features

Binary Category Features	Numeric Features for Naïve Bayes probability calculation
Presence of JS, Presence of NS, Presence of external JS, Presence of image, Presence of iframe, Presence of links, Presence of Country Traffic rank, Presence of World Traffic rank, Presence of fake form	Number of SMS API call, Number of tel API call, Number of APK API call, Number of JS, Number of NS, Number of external JS, Number of internal JS, Number of image, Number of iframe, Number of redirects, Number of links, Length of URL, Number of forward slash, Number of Question marks, Number of dots, Number of hyphens, Number of underscore, Number of equal signs, Number of ampersand sign, Number of digits, world traffic rank, country traffic rank, and Number of fake form

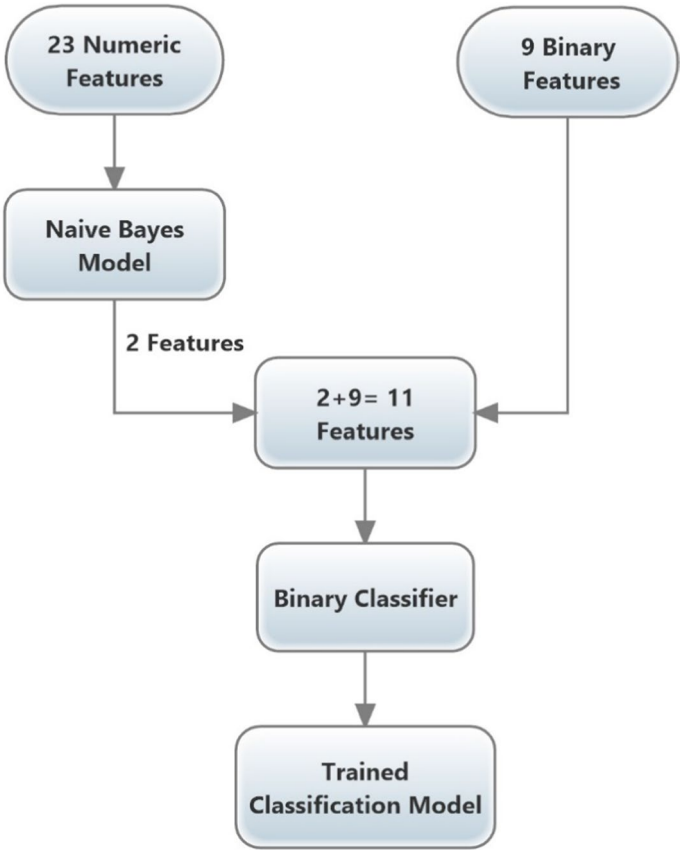

Fig. 5 Feature selection model

feature. MI is symmetric and non-negative and is equal to zero if and only if two random variables are independent, and higher values mean higher dependency. In MI, the feature gives non-zero value if it is relevant in correct classification. From Fig. 6, we have observed that most of the features are strong indicators for our model.

5.3 Results on Various Classifiers

Various classification algorithms have been applied and checked against the test set. The different algorithms machine learning algorithms are Logistic regression, K nearest neighbors, support vector machine, decision tree and random forest classification. We have used 70% of data for training purpose, and the remaining 30% data for testing purpose. We have calculated the true positive rate, true negative rate, accuracy, precision and f1 score for all the algorithms as shown in Table 4 [16]. Table 5 presents the TPR, TNR, Accuracy and Precision, the experiment results of our method on various classifiers. From Table 5, we have determined that Random Forest Classifier works best for our model.

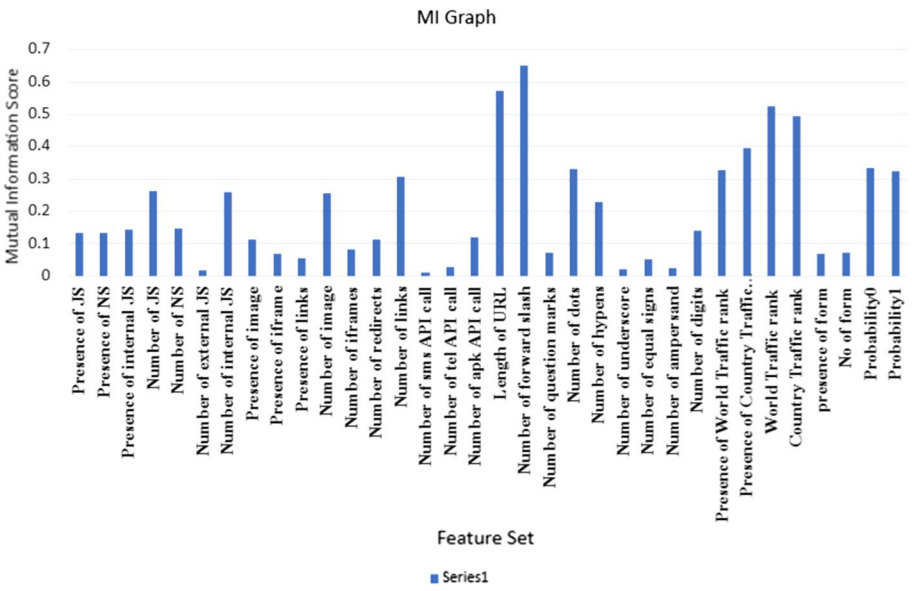

Fig. 6 Representation of Feature Importance

5.4 Evaluation of Features

In this experiment, we have evaluated the efficiency of each category of the proposed feature set. We have analyzed the TPR, TFR and accuracy of our model by selecting different categories of the features. Table 6 presents the results of the proposed approach by combining all categories of features. It is observed that all the features Mobile-specific, JavaScript, HTML, URL, Website specific, Fake form Naive Bayes are important for detection of phishing websites. From our analysis, we found that none of the features alone able to get accuracy near to our final model except website-specific features which based on the rank of URL from alexa.com. However, for newly created webpages, this feature gives false result [21]. Therefore, we have considered all the features to improve the detection accuracy of the proposed mobile phishing detection model.

5.5 Comparison with Existing Mobile Webpage Detection Techniques

We have considered three approaches for this comparison. The comparison is based on the type of approach, advantages, limitations, TPR, FPR, and accuracy as presented in Table 7. The first is KAYO [7], which is a technique to detect malicious mobile webpages using static analysis. In this approach, a high number of features (i.e. 44) are used for the detection in which some of them do not contribute to the accuracy notably. Moreover, this model divided the URLs into two categories namely mobile version and desktop version. If any URL fall in the desktop version category then the model uses Google safe browsing API. However, most of the URLs belonging to the desktop category and only a few URLs detected by the machine learning model. Moreover, the detection accuracy of our approach is better than KAYO. Johny et al. [11] classified mobile webpages

Table 4 Performance measures used in our approach

Measure	Formula	Description
TPR	$\text{TPR} = \frac{N_{P \to P}}{N_P} \times 100$	Rate of phishing websites classified as phishing out of total phishing websites
TNR	$\text{TNR} = \frac{N_{L \to L}}{N_L} \times 100$	Rate of legitimate websites classified as legitimate out of total legitimate websites
Accuracy	$\text{Accuracy} = \frac{N_{L \to L} + N_{P \to P}}{N_L + N_P} \times 100$	The rate of phishing and legitimate websites which are identified correctly with respect to all the websites
Precision	$\text{Precision} = \frac{N_{P \to P}}{N_{P \to P} + N_{L \to P}} \times 100$	Measures the rate of instances correctly detected as phishing with respect to all instances detected as phishing

Table 5 Results of proposed approach on various classifiers

Classification Algorithm	TPR	TNR	Accuracy	Precision (%)
Logistic Regression	91.4754%	92.6229%	92.0491%	92.5373
K-Nearest Neighbors	88.2258%	95.8333%	91.9672%	95.6294
Support Vector Machine	86.7742%	95.5%	91.0656%	95.2212
Random Forest	93.225806	94.5	93.852459	94.5990
Decision Tree	92.9032%	94.3333%	93.6066%	94.4262
Neural Network	93.3871%	94.1667%	93.7705%	94.2997

Table 6 Performance of Proposed Approach on different combination of features on Random Forest classifier

Features	TPR (%)	TNR (%)	Accuracy (%)
JavaScript	88.96	56.87	74.66
HTML	67.77	81.59	73.93
Website Specific	92.27	92.03	92.41
Fake form	70.65	62.50	66.64
Naive Bayes	95.00	80.50	87.87
JavaScript + HTML + URL + Website Specific + Fake form + Naive Bayes	93.23	94.50	93.85

based on matching the content of the webpage. However, this approach not considered efficient features, which are recently used for phishing attack such as JavaScript, and HTML features. Moreover, if an attacker creates a fake webpage that looks like a genuine one, then this approach fails to detect such webpages.

5.6 Advantages of Our approach

5.6.1 Language Independence

The language dependency for the detection of malicious website is a problem for most of the existing approaches. Only 25.23% of websites are in the English language [22]. Therefore, it is very much important for detecting website language independently. In our approach, all the features are independent of the textual content of the website. Therefore, our approach produces result independent of the textual content of the website.

5.6.2 Drive by Downloads Attack Detection

A drive-by download attack refers to the unintentional download of malicious code on the system. Malicious JavaScript is used to performed drive-by download attack. Our approach can detect drive-by download attack because it uses JavaScript and HTML features [23].

Table 7 Comparisons with existing mobile phishing detection techniques

Approach	Type	Advantages	Limitations	Accuracy	TPR	FPR
KAYO [7]	ML	Use static feature	Use of some and ineffective features	90%	89%	8%
MP-shield [10]	ML	ensuring high level of security even with unsecured browsers	Low accuracy	89.2%	89.2%	14.4%
Johny et al. [11]	content matching	Able to detect zero day attacks	If attacker create exactly same looking webpage then it unable to detect fake webpage	92%	–	–
Proposed Approach (APuML)	ML	(1) Most of them are static feature (2) High detection accuracy (3)Low response time	Depends on third party for finding rank of the website	93.85%	93.23%	5.5%

5.6.3 Zero-day Attack Detection

Blacklist/White-list and visual similarity-based based approaches cannot detect zero-day attacks [16]. On the other hand, machine learning approaches can detect zero-day attack [24].

5.6.4 Clickjacking Attack Detection

Clickjacking is an attack that tricks a user into clicking a webpage element that is invisible or disguised as another element. This can cause users to unwittingly download malware, visit malicious webpages, provide credentials or sensitive information, transfer money, or purchase products online. Typically, clickjacking is performed by displaying an invisible page or HTML element, inside an iframe, on top of the page the user sees. As our approach that includes HTML features can detect clickjacking attack with high accuracy [16].

5.6.5 Prominent Features

Our features set is carefully taken to get the maximum correct classification result for legitimate and phishing websites. Analyzing MI Graph for all the features, we get to know that all the features used in our model are prominent as shown in Fig. 4, where a higher value indicates more important features for classification. We experimentally found that some of the features that are currently absolute such as mms, mmsto, International Phonetic Alphabet (IPA) are removed from our features set which are used by the previous literature [7].

5.6.6 Low Response Time

Our proposed method for detecting phishing webpages has an average response time of 4.0955 s, which is amazingly fast as compared to other machine learning methods, which have a response time of 10–13 s. The time it takes to get to the source code and produce a result is negligible.

5.7 Discussion

Mobile version of websites are different from the desktop version of websites based on functionality, content, and layout. Therefore, existing tools, which used static features to detect malicious desktop version of webpages, do not properly work for the mobile version of webpages. Following are some of the differences between the mobile version of the websites and the desktop version of the websites:

i. *Differences in content*: Mobile version of websites are simpler than their desktop version of websites. Therefore, the value of particular static features such as the number of JavaScript on the mobile version of webpages differs from that of the desktop version of webpages. Approximately 90% of the mobile version of webpages does not have any iframes [7]. The desktop version of webpages has more JavaScript than the mobile version of webpages. Similarly, the number of images in the mobile versions are different from the desktop versions of the websites.
ii. *Infrastructure*: The number of redirections in the mobile version of the webpages are greater than the desktop version of the webpages. Such behaviour suspects malicious

activity in the desktop version of the website [16]. However, multiple redirects do not consider malicious activity in the mobile version of webpages due to the characteristics of hosting infrastructure.

iii. *Mobile-specific functionality*: Mobile version of webpage can access a user's personal information by using different mobile-specific web APIs [8]. Existing static analysis method/tools do not consider such mobile-specific functionalities in their feature set. Such features are very much helpful for the detection of malicious webpage.

iv. *Lightweight webpage*: Mobile webpages are generally lightweight compare to that of the desktop version of the webpages due to the low processing power of mobile devices [9].

5.8 Limitations

The concerns with our system are similar to the concerns with other existing approaches. The attackers can evade by imitating the features that are considered as strong indicators for the legitimacy of the website. However, our thorough set of features make that difficult. we could not validate the phone number given during the API calls to the dialer or SMS application, etc. The threats on the mobile web will potentially increase in the upcoming years and so our system needs to be updated to detect those threats.

6 Conclusion and Future Work

Mobile webpages are completely different from desktop ones in terms of functionality, layout and content but the existing solutions either are completely based on data feed or ML based solution that use many static features which are not effective and efficient in real time. In our approach, we determined the legitimacy of a webpage through various stages, including data feed, DNS and machine learning. Therefore, our approach can prove to be an effective solution to detect malicious phishing sites, click hijacking, drive by downloads, forgery malware and also detect zero-day attacks. Moreover, the selection of a new feature set for improving detection accuracy is one of the main contributions of our work. In future, we will try to scale our model to a broader audience and make it convenient for the users. This is possible only by making the model automatic and integrating the detection system on web browsers regularly used by users in mobile devices like "Chrome", "UC browser", etc.

Funding Not applicalble.

Data Availability The datasets generated and analyzed during the current study will be available on request.

Declarations

Conficts of interest The authors declare that they have no confict of interest.

Consent for publication Yes.

References

1. "Percentage of Internet Traffic is Mobile?," Available: https://www.oberlo.in/statistics/mobile-internet-traffic (Last Accessed on 7th June 2021).
2. S. O'Dea, "Number of smartphones sold to end users worldwide from 2007 to 2020*(in million units),"* Available: https://www.statista.com/statistics/263437/global-smartphone-sales-to-end-userssince-2007/ (last accessed: 26 May 2021).
3. Crane, C. "20 Phishing Statistics to Keep You from Getting Hooked in 2019," Available: https://www.thesslstore.com/blog/20-phishing-statistics-to-keep-you-from-getting-hooked-in-2019/ (Last accessed on 26 May 2021).
4. Arghire, I. "Phishers Serve Fake Login Pages via Google Translate," Available: https://www.securityweek.com/phishers-serve-fake-login-pages-google-translate (Last accessed on 26 May 2021).
5. "Beware of criminals pretending to be WHO", Available: https://www.who.int/about/communications/cyber-security (Last accessed on 26 May 2021).
6. Crane, C. "Coronavirus Scams: Phishing Websites & Emails Target Unsuspecting Users," Available: https://www.thesslstore.com/blog/coronavirus-scams-phishing-websites-emails-target-unsuspecting-users/ 2020, (Last accessed on 26 May 2021).
7. Amrutkar, C., Kim, Y. S., & Traynor, P. (2016). Detecting mobile malicious webpages in real time. *IEEE Transactions on Mobile Computing, 16*(8), 2184–2197.
8. Goel, D., & Jain, A. K. (2018). Mobile phishing attacks and defence mechanisms: State of art and open research challenges. *Computers & Security, 73*, 519–544.
9. Rao, R. S., Vaishnavi, T., & Pais, A. R. (2019). PhishDump: A multi-model ensemble based technique for the detection of phishing sites in mobile devices. *Pervasive and Mobile Computing, 60*, 101084.
10. Bottazzi, G., Casalicchio, E., Cingolani, D., Marturana, F., Piu, M. (2015). MP-shield: A framework for phishing detection in mobile devices. In *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, 2015, pp. 1977–1983, IEEE.
11 Johny, C. N., & Ratheesh, T. K. (2018). Novel Defence Scheme for Phishing Attacks in Mobile Phones. *International Conference on Intelligent Data Communication Technologies and Internet of Things* (pp. 1174–1181). Cham: Springer.
12. "List of phishing websites", Available: www.phishtank.com (Last accessed on 26 May 2021).
13. Li, L., Berki, E., Helenius, M., & Ovaska, S. (2014). Towards a contingency approach with whitelist- and blacklist-based anti-phishing applications: What do usability tests indicate? *Behaviour & Information Technology, 33*(11), 1136–1147.
14 Jain, A. K., & Gupta, B. B. (2017). Phishing detection: analysis of visual similarity based approaches. *Security and Communication Networks, 2017*, 1–20. https://doi.org/10.1155/2017/5421046
15. Gastellier-Prevost, S., Granadillo G.G., Laurent, M. (2011). A dual approach to detect pharming attacks at the client-side. In *2011 4th IFIP International Conference on New Technologies, Mobility and Security*, 2011, pp. 1–5, IEEE.
16. Jain, A. K., & Gupta, B. B. (2019). A machine learning based approach for phishing detection using hyperlinks information. *Journal of Ambient Intelligence and Humanized Computing, 10*(5), 2015–2028.
17. "List of top legitimate websites", Available: www.alexa.com (Last accessed on 26 May 2021).
18. "List of phishing websites", Available: www.openphish.com (Last accessed on 26 May 2021).
19. I. Rish (2001). An empirical study of the naive Bayes classifier. *IJCAI 2001 workshop on empirical methods in artificial intelligence*, *3*(22), 41–46.
20. Vergara, J. R., & Estévez, P. A. (2014). A review of feature selection methods based on mutual information. *Neural computing and applications, 24*, 175–186.
21. Jain, A. K., & Gupta, B. B. (2018). Two-level authentication approach to protect from phishing attacks in real time. *Journal of Ambient Intelligence and Humanized Computing, 9*(6), 1783–1796.
22. "Internet World Users by Language", Available: https://www.internetworldstats.com/stats7.htm (Last accessed on 26 May 2021).
23 Aldwairi, M., Hasan, M., & Balbahaith, Z. (2020). Detection of drive-by download attacks using machine learning approach. *Cognitive Analytics: Concepts Methodologies Tools and Applications* (pp. 1598–1611). IGI Global.
24. "The Performance of Machine and Deep Learning Classifiers in Detecting Zero-Day Vulnerabilities", Available: https://deepai.org/publication/the-performance-of-machine-and-deep-learning-classifiers-in-detecting-zero-day-vulnerabilities (Last accessed on 26 May 2021).

Dr. Ankit Kumar Jain is presently working as Assistant Professor in National Institute of Technology, Kurukshetra, since September 2013. He received Master of technology from Indian Institute of Information Technology Allahabad (IIIT) India . Dr. Jain received PhD degree from National Institute of Technology, Kurukshetra in the area of Information and Cyber Security. He has more than 50 research papers in International journals and conferences of high repute including Elsevier, Springer, Taylor & Francis, Inderscience, IEEE, etc. His general research interest is in the area of Information and Cyber security, Phishing Website Detection, Web security, Mobile Security, IoT Security, Online Social Networks and Machine Learning.

Ninmoy Debnath has received his B.Tech. degree in Computer Science and Engineering from Central Institute of Technology, Kokrajhar, India. Currently, he is pursuing M.Tech. in Computer Engineering from National Institute of Technology, Kurukshetra, Haryana, India. His research interest includes Mobile security, Web security, Machine learning and phishing detection.

Dr. Arvind Kumar Jain is presently working as Associate Professor in National Institute of Technology Agartala, India. He received Master of technology and Ph.D. degree from Indian Institute of Technology Kanpur (IIT-K) India. He received POSOCO Power System Award-2014 for doctoral research work. His general research interest is in the area of Electric mobility, Application of evolutionary techniques and machine learning.