HR et al. Cybersecurity (2020) 3:20
https://doi.org/10.1186/s42400-020-00059-1

Cybersecurity

RESEARCH Open Access

Development of anti-phishing browser based on random forest and rule of extraction framework

Mohith Gowda HR[1*], Adithya MV[2], Gunesh Prasad S[3] and Vinay S[4]

Abstract

Phishing is a technique under Social Engineering attacks which is most widely used to get user sensitive information, such as login credentials and credit and debit card information, etc. It is carried out by a person masquerading as an authentic individual. To protect web users from these attacks, various anti-phishing techniques are developed, but they fail to protect the user from these attacks in various ways. In this paper, we propose a novel technique to identify phishing websites effortlessly on the client side by proposing a novel browser architecture. In this system, we use the rule of extraction framework to extract the properties or features of a website using the URL only. This list consists of 30 different properties of a URL, which will later be used by the Random Forest Classification machine learning model to detect the authenticity of the website. A dataset consisting of 11,055 tuples is used to train the model. These processes are carried out on the client-side with the help of a redesigned browser architecture. Today Researches have come up with machine learning frameworks to detect phishing sites, but they are not in a state to be used by individuals having no technical knowledge. To make sure that these tools are accessible to every individual, we have improvised and introduced detection methods into the browser architecture named as 'Embedded Phishing Detection Browser' (EPDB), which is a novel method to preserve the existing user experience while improving the security. The newly designed browser architecture introduces a special segment to perform phishing detection operations in real-time. We have prototyped this technique to ensure maximum security, better accuracy of 99.36% in the identification of phishing websites in real-time.

Keywords: Phishing attack, Machine learning, Intelligent browser engine, Rule of extraction algorithm, Browser architecture

Introduction

The Internet has widely spread all over the world covering every field of work. As a result, users who depend on the internet to carry out their businesses are also increasing considerably. This number tempts the imposters to carry out their fake operations. Eventually, end-users become more vulnerable to various kinds of web-attacks. One of the major implications of these web attacks affects the financial transactions over the internet (Phishing Trends and Intelligence Report 2018 n.d.). Phishing is one amongst the popular techniques that is used to gain the advantage of such security flaws. It is a cyberattack that is described as the art of mimicking a legitimate website of an authentic business targeting to gain access over its secretive information. These websites have extremely high graphical similarities to the real ones (Jain and Gupta 2017). Normally, these attacks are carried out by sending a website that is exactly similar to the real one to the victim asking

* Correspondence: mmohithgowda@gmail.com
[1]B.E in Computer Science and Engineering, PES College of Engineering, 4011, Vasuda Krupa, 3rd Cross, Shankar Nagar, Mandya, Karnataka 571401, India
Full list of author information is available at the end of the article

him to update his information. Detecting and blocking a phishing attack is extremely important to preserve the security and confidentiality of an individual over the internet. Researches have come up with various approaches (Armano et al. 2016; Hu et al. 2016; Ma et al. 2009; Roy et al. 2013; Sahingoz et al. 2019; Williams and Li 2017) to solve this prominent problem. However, they fail in some way to be easily used by every individual. To cite an example, there are several machine learning algorithms developed to detect phishing sites. But these can only be used by a technical user. Yet another example is that the researchers have come up with phishing detection website to check website authenticity. The downside is that this being a manual process and the users cannot verify for all the websites that he visits. Even extensions are not efficient and they lack in accuracy and speed.

The main objective of this paper is to develop a technique that can be easily used by everyone to detect non-legitimate websites accurately in real-time. The detection process is carried out on the client-side with less processing. The novelty of EPDB approach is the newly designed browser architecture which is built by modifying existing browser architecture to introduce a new module named "Intelligent Engine" that is responsible for the easy detection of phishing websites in real-time. This module consists of Random Forest Classification and Rule of Extraction Framework. The Rule of Extraction algorithm uses 30 different features to analyze a website with only the URL entered by the user. Then the result of this is used by the Random Forest Classification algorithm to determine its authenticity. The classification model is trained by the dataset, consisting of 11,055 illegitimate URLs. The Intelligent Engine analyses every website that is loaded by the browser. The Intelligent Engine and Rendering Engine are designed to work in such a way that, they execute in parallel to minimize time. With 30 different features for analyzing the URLs, a variety of URLs can be detected. The classification model ensures better accuracy in the identification of phishing websites. The Intelligence Engine module reduces time taken in the detection of phishing websites. Overall, EPDB technique has proven to detect newly generated URLs in real-time with 99.36% accuracy.

The remaining paper is structured as follows: Literature Review is covered in Section II, followed by System Analysis in Section III. System Model comes next i.e. in Section IV. In Section V we will introduce our technique in the detection of phishing websites. We Evaluate the real-world performance of the proposed EPDB model with a comparison with existing approaches in Section VI. Lastly, the Final Remarks with future enhancement is covered in Section VII.

Literature review

This section covers literature review on various relevant works.

Prominent existing approaches for detection of phishing websites can be categorized as follows:

Detect and block the phishing web sites manually in time

Detecting phishing webpages manually is one of the common approaches. User needs to be aware of various kinds of phishing-attacks and prior knowledge is essential in identifying these webpages in real-time. Williams and Li (2017) proposed an architectural model that evaluates ACT-R cognitive behavior. This is carried out by analyzing the authenticity of webpages based on the HTTP padlock security indicator. Afroz and Greenstadt (2011) has come up with a technique called 'PhishZoo' which uses site profiling as well as profile matching in the detection process. This technique makes a list of all sensitive websites and this list will be used to compare the loaded website. This approach is mainly based on matching the content of the Legitimate webpage with the Non-Legitimate one.

Detection based on URL and content of websites

Detection methods based on URL uses various characteristics of the website URL to filter phishing websites. Ma et al. (2009) implements learning online along with methods to identify host-based and lexical properties of phishing website URLs.

Content-based detection compares the content of the website viewed by the user with the original one. Mao et al. (2017) have proposed a system that detects phishing by analyzing similarities in components in websites. This method uses URL tokens to improve prediction accuracy of illegitimate websites. In addition to that, it compares the CSS rules of the legitimate and non-legitimate websites to identify the phishing one. Futai et al. (2016) uses the Graph Mining technique to detect phishing webpages. This method detects those phishing websites that aren't possible by the URL analysis technique. It also accounts for the repeated interaction between the website and the user. Therefore, by analyzing the statistics of repeated interaction between the website and the user, it generates the AD-URL graph which is used to detect the phishing website.

Block the phishing e-mails by various spam filter software

Email attacks are a major source leading user to phishing websites. Spam filters are great options to prevent spam email clicks. Spam filters ensure a wide majority of malicious spam emails detection and are not delivered to inboxes. Roy et al. (2013) has developed a technique that uses spam filters to detect spam emails. This uses the Naive Bayes Classifier model for the

prediction. It classifies by analyzing the contents in legitimate and illegitimate mails. It has managed to have an accuracy of 85%. Pandey and Ravi (2013) has come up with a technique where they use the URL and the source code of the website to gather information on the dissimilarities. They perform a text analysis on the gathered information and finally make a prediction.

Server-side detection

Hu et al. (2016) has proposed a technique that analyzes server log information to identify phishing websites. When a user visits an illegitimate webpage, the browser contacts the real one for resources. This request is registered in the log by the legitimate website server, later this is used to identify illegitimate ones. Wu et al. (2019) has come up with a technique that uses fuzzy logic combined with the power of machine learning and eliminating the use of Boolean algorithm in the system. They make use of domain name, sub-domain name and also the lifetime of the webpage in the authentication process.

Client-side detection

Anti-phishing software contains a computer code that identifies phishing websites and other forms used to access the data. These tend to block the content usually with a warning to the user. Anti-virus and Anti-malware are software's that falls into this category. Armano et al. (2016) has proposed a real-time method to detect phishing websites by developing an add-on or extension for a browser. It extracts information from the websites visited by the user to identify a phishing website, then a caution message is popped on the screen if the website is phishing. Marchal et al. (2017) has proposed a similar kind of real-time browser extension for the Firefox browser.

Other detection methods

Mei et al. (2016) proposed a technique that gets features from the website and with the help of the support vector machine classifier model, the prediction on the authenticity of the website is made. Here, the model is trained first and then it is tested on various test cases.

Hawanna et al. (2016) has proposed a system that uses a novel algorithm to detect phishing websites. It considers various test methods like Alexa ranking, blacklist search, to detect phishing websites. It works well for websites with HTTP protocols. Sahingoz et al. (2019) has used several classification algorithms with NLP to detect phishing websites in real-time. It has shown accuracy of about 97.98%.

Major disadvantages of all the methods listed above are listed in the Table 1.

System analysis

Problem statement

Criminals use phishing attacks to steal user credentials to obtain access to user's private data. According to the Federal Bureau of Investigation (FBI)'s (2017 Internet Crime Report n.d.) report, the total number of phishing scams detected in 2017 is 25,344 incurring an overall loss of about $29,703,421. Fields that are most affected by phishing are Payment, Financial Institution, Webmail, Cloud Storage/Hosting, commerce/Retail, Telecom,

Table 1 Downsides of the existing phishing detection methods

Paper	Method	Disadvantages
Williams and Li (2017), Afroz and Greenstadt (2011)	Detect and block the phishing web sites manually in time.	• Most of the internet users do not have the knowledge to identify a phishing webpage in real-time. • Even trained people fall into the attack because people tend to forget to check the website's legitimacy while they are busy with their work. • Security awareness training is not continuous.
Ma et al. (2009), Mao et al. (2017), Futai et al. (2016)	Detection based on URL and Content of Websites.	• They lack in new website URL detection • These methods are not accurate and they tend to modest false-negative rate.
Roy et al. (2013), Pandey and Ravi (2013)	Block the phishing E-mails by various spam filter software	• These spam filters tend to block genuine messages. • They fail to detect these attacks apart from email-threads.
Hu et al. (2016), Wu et al. (2019)	Server-side Detection	• Users will receive delayed responses from servers about the authenticity of the website. • They underperform in slow internet connections.
Armano et al. (2016), Marchal et al. (2017)	Client-side Detection	• These software's signature-based security controls are proving less and less effective as years pass by. For example, these solutions are not particularly good at identifying file-less malware. • They utilize a lot of memory.
Mei et al. (2016)	Other Detection methods	• It is not effective on pages that are not visited previously and websites should be maintained by constantly updating to preserve better accuracy.

Social Media. These are the main fields where the phishing has affected the most. As of phish labs reporting (Phishing Trends and Intelligence Report 2018 n.d.) 2017, over 26% of all phishing attacks target the Email/Online Services, over 20% of phishing attacks were made on the financial sector, and around 16% targeted the Payment Services. According to APWGs Phishing Activity Trends Report released each quarter 2019 (APWG trends report q1 2019 n.d.; APWG trends report q2 2019 n.d.; APWG trends report q3 2019 PRODUCTION n.d.; APWG trends report q4 2019 n.d.), the total number of cyber-crimes via phishing webpages are dramatically increasing in a very huge number. It is said that the second quarter has a greater number than the first quarter of 2019, and it is also much greater than the second half of the year 2018. Considering the statistical data released by APWGs quarterly, we have pictured the total number of phishing sites that were detected every month in the year 2019 in Fig. 1. According to APWG, the most targeted sectors through phishing attacks in 2019 is the SAAS/Webmail which is 34%, then comes the Payment with 23%, Financial Institutions stand third with 18% as depicted in the Fig. 2. To control this most of the companies are investing a huge amount of money on security, on average, it is 11.7 million USD.

System model

System architecture

The proposed system architecture of EPDB is illustrated in Fig. 3. It is designed to perform all the operation that a browser needs, along with this, a new module named as "Intelligent Engine" is introduced to perform operations to detect phishing websites while surfing the web.

The main components of the Browser are as follows:

- User Interface:

It offers a means by which, a user can interact with the Browser Engine. It incorporates various functionalities such as the address box, navigation button, bookmarks, favorites, etc. User Interface is that part of the browser that is displayed, apart from the window where the webpage is displayed.

- Browser Engine:

Browser engine comes in between UI and rendering engine and it ensures a high-level interface to the rendering engine. It offers several features like loading the website and navigating through it. It also provides several error messages that occur due to loading.

- Rendering Engine:

The rendering engine is responsible for converting the URL to its graphical form. Basically, it is an interpreter, that interprets the webpage that is comprised of HTML, XML, CSS, etc. The core of the rendering engine is HTML parser which is responsible for parsing HTML

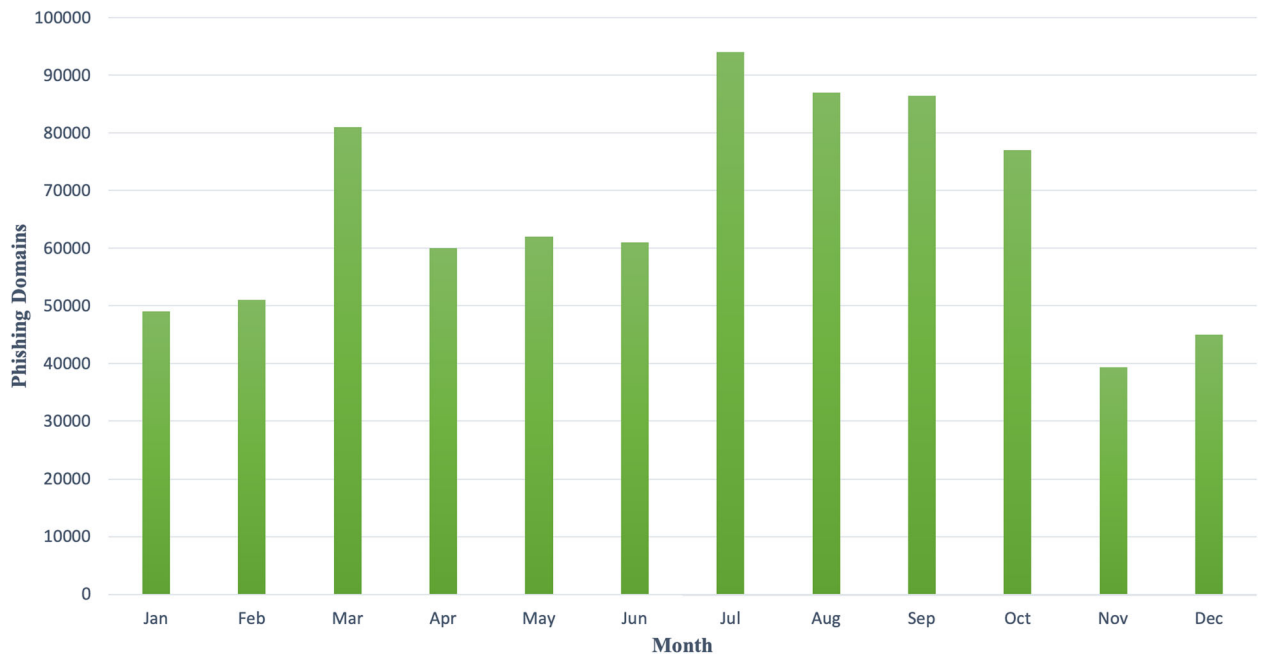

Fig. 1 Phishing Sites 2019 (APWG trends report q1 2019 n.d.; APWG trends report q2 2019 n.d.; APWG trends report q3 2019 PRODUCTION n.d.; APWG trends report q4 2019 n.d.)

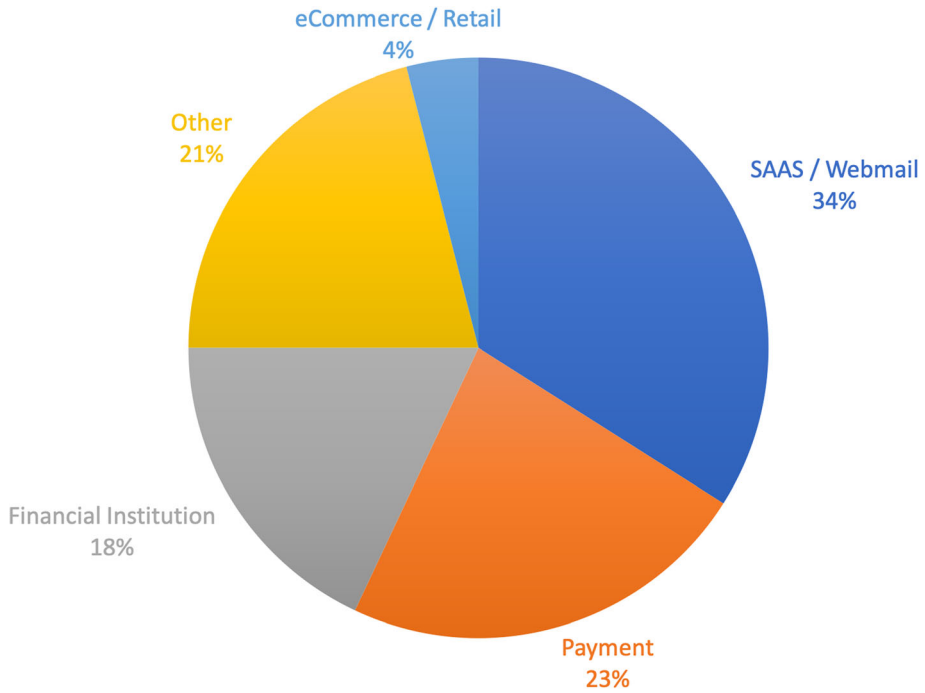

Fig. 2 Most-Targeted Industry Sectors 2019 (APWG trends report q1 2019 n.d.; APWG trends report q2 2019 n.d.; APWG trends report q3 2019 PRODUCTION n.d.; APWG trends report q4 2019 n.d.)

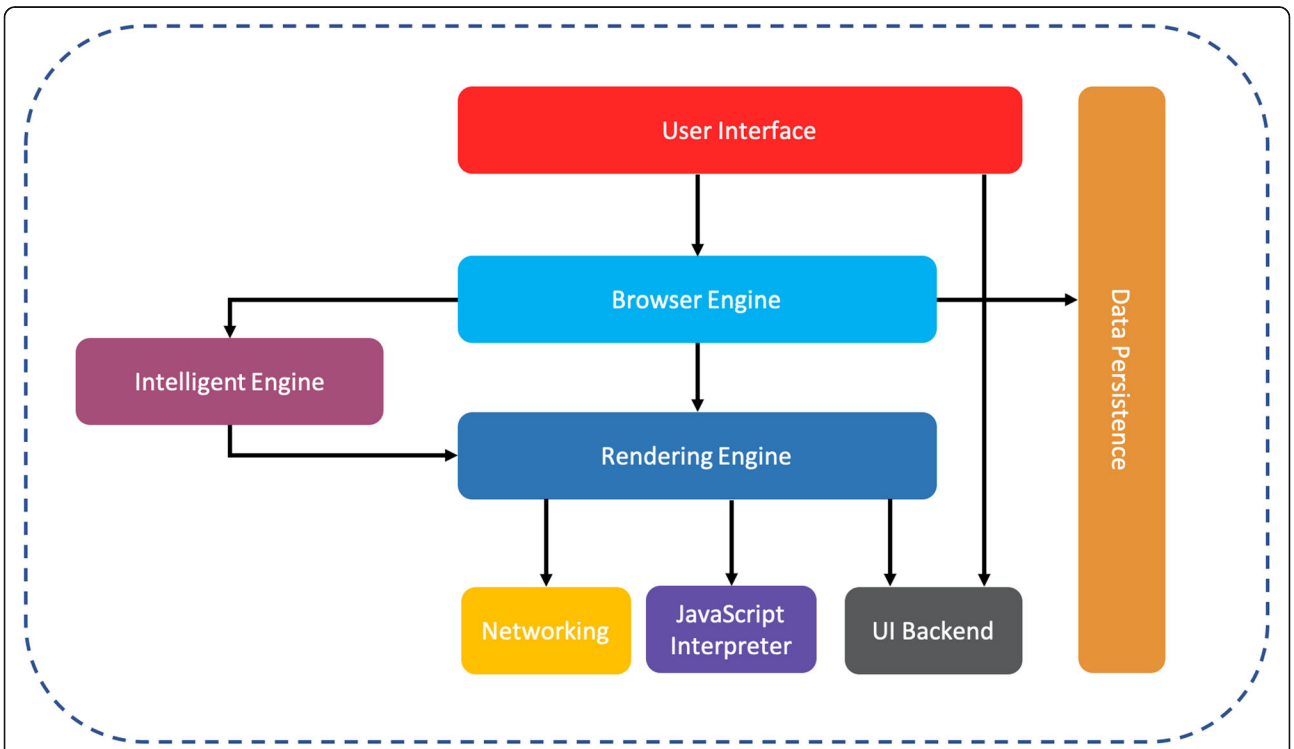

Fig. 3 Proposed architecture of Embedded Phishing Detection Browser (EPDB)

contents. Overall it generates a layout to be viewed in the user interface.

- Networking:

This uses various protocols like HTTP, HTTPs, FTP, etc., to fetch the website through the URL requested by the user. It is also responsible for providing security to the user, and establish a secure internet connection, maintain and close communication between two end-users on the internet. It provides features to cache frequently visited websites to reduce network traffic.

- JavaScript Interpreter:

JavaScript interpreter interprets the JavaScript code that comes along as a part of the webpage and passes the results for rendering. It provides functionalities such that it provides several options to develop a responsive, interactive webpage.

- UI Backend:

It invokes operating system methods to create windows, widgets and other things related to graphics.

- Data Storage:

It provides a web database feature to store webpages for reading mode, bookmarks, settings, cookies, etc.

- Intelligent Engine:

This section is responsible for the detection of phishing websites in real-time. It uses the rule of extraction framework and random forest classifier algorithm to identify a webpage legitimacy. It takes the URL from the browser engine, verifies it and finally it will send a message to the rendering engine. If the message says the website is not legitimate, then the rendering engine popup an alert to the user and providing options to the user to either go back to safety or continue. The overall process of the Intelligent engine is completed before the rendering engine renders the webpage. This engine carefully examines every webpage the user visits while browsing through the web.

Proposed EPDB scheme

Overview on the dataset

The proposed EPDB work gathered phishing websites from phish tanks (Join the fight against phishing n.d.) and millers' miles (Phishing scams and spoof emails at MillerSmiles.co.uk n.d.). The collection consists of 11, 055 records comprising of both Legitimate and Illegitimate websites. The exact count of both the category present in the dataset is shown in Fig. 4. Every tuple in

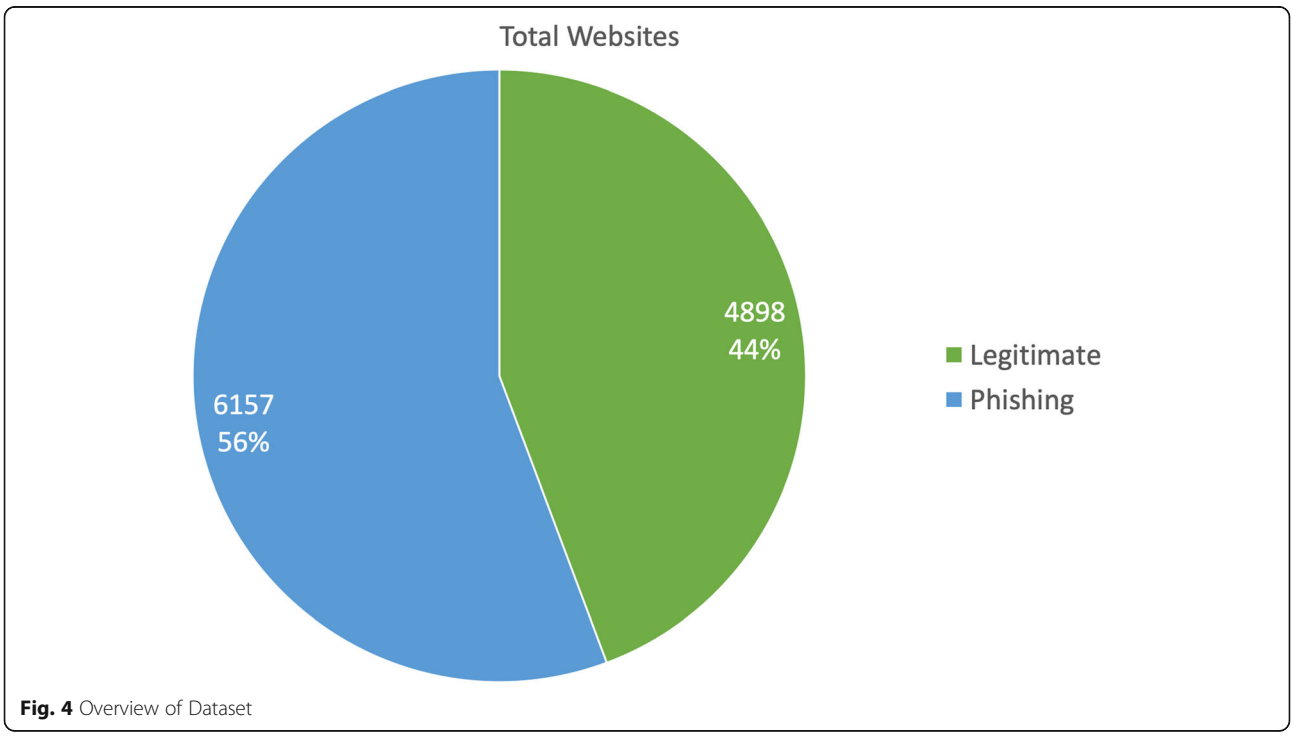

Fig. 4 Overview of Dataset

Table 2 A list of 30 distinct features

S.No	Features	Conditions	Result
1.	IP Address	Domain Part of URL contains IP Address	Phishing
		Domain Part of URL doesn't contain IP Address	Legitimate
2.	Length of URL	Length of URL below 54	Legitimate
		Length of URL greater or equal to 54	Suspicious
		Length of URL lesser or equal to 75	Phishing
3.	Shortening Services	Very Short URL	Phishing
		Normal URL	Legitimate
4.	'@' Symbol	Existence of '@' character in URL	Phishing
		Absence of '@' character in URL	Legitimate
5.	Double slash forwarding	Position of Last '// in URL is below 7	Phishing
		Position of Last '// in URL is above 7	Legitimate
6.	Prefix and Suffixes	Existence of '–' character in Domain name	Phishing
		Absence of '–' character in Domain name	Legitimate
7.	Sub Domain	No. of Dots equal to one in Domain Part of URL	Legitimate
		No. of Dots equal to two in Domain Part of URL	Suspicious
		No. of Dots greater than two in Domain Part of URL	Phishing
8.	SSL final Certificate	Using https by Trusted providers and Certificate Age should be greater than or equal to 1 Year	Legitimate
		Using https with Non-Trusted providers	Suspicious
		Using https by Non-Trusted providers and Certificate Age lesser than to 1 Year	Phishing
9.	Domain registration length	Expiry date of Domains lesser or equal to 1 year	Phishing
		Expiry date of Domains greater than 1 year	Legitimate
10.	Favicon	Favicon retrieved from External source	Phishing
		Favicon retrieved from Internal source	Legitimate
11.	Non-Standard Ports	Port No. has Preferred Status	Phishing
		Port No. doesn't have Preferred Status	Legitimate
12.	"HTTPS" token	Domain section with HTTP token	Phishing
		Domain section without HTTP token	Legitimate
13.	URL Requests	Percent of request URL lesser than 22%	Legitimate
		Percent of request URL is greater than or equal to 22% and lesser than 61%	Suspicious
		Percent of request URL is greater than 61%	Phishing
14.	URL with anchor	Percent of request URL lesser than 31%	Legitimate
		Percent of request URL is greater than or equal to 31% and lesser than 67%	Suspicious
		Percent of request URL is greater than 67%	Phishing
15.	Tags containing Links	Percent of Links in "Meta"," Link" and "Script lesser than 17%	Legitimate
		Percent of Links in "Meta"," Link" and "Script" is greater than or equal to 17% and lesser than 81%	Suspicious
		Percent of Links in "Meta"," Link" and "Script" is greater than 81%	Phishing
16.	Server Form Handler-SFH	"Is Empty" or "about: blank" in SFH	Phishing
		SFH forwards to another Domain	Suspicious
		SFH doesn't contain "Is Empty" or "about: blank" or doesn't forwards to another domain	Legitimate
17.	Submitting to email	"mail()" services usage	Phishing
		Non-usage of "mail()"	Legitimate
18.	Abnormal URL	URL without Hostname	Phishing
		URL with Hostname	Legitimate
19.	Webpage Redirect	Page redirect is lesser than or equal to one	Phishing
		Page redirect is greater than or equal to two and less than four	Suspicious
		Page redirect is greater than four	Legitimate
20.	On mouse over	Change in status bar with mouse over	Phishing
		No Change in status bar with mouse over	Legitimate
21.	Mouse right clicks	Disabled Right Click	Phishing
		Enabled Right Click	Legitimate

Table 2 A list of 30 distinct features *(Continued)*

S.No	Features	Conditions	Result
22.	Browser Pop up	Browser Popups with text boxes	Phishing
		Browser Popups without text boxes	Legitimate
23.	Iframe	Webpage with usage of iframe	Phishing
		Webpage without the use of iframe	Legitimate
24.	Age of domain	Domain age greater than 6 months	Phishing
		Domain age lesser than 6 months	Legitimate
25.	DNS Record	Domain without DNS record	Phishing
		Domain with DNS record	Legitimate
26.	Web traffic	webpage rank less than or equal to 100,00	Legitimate
		webpage rank greater than 100,00	Suspicious
		webpage rank greater than 100,000	Phishing
27.	Page Rank	Page Rank less than 0.2	Phishing
		Page Rank greater than 0.2	Legitimate
28.	Google Index	Webpage without google index	Phishing
		Webpage with google index	Legitimate
29.	Links pointing to page	No. of Links Pointing to Webpage is zero	Phishing
		No. of Links Pointing to Webpage is less than or equal to two	Suspicious
		No. of Links Pointing to Webpage is greater than two	Legitimate
30.	Statistical analysis report	Host having topmost Phishing IP Addresses	Phishing
		Host without topmost Phishing IP Addresses	Legitimate

the dataset possesses 30 different characteristics that a website will have. These characteristics will be considered as the independent variables for training the model. Based on these features one dependent variable or target function is defined, which defines the authenticity of the website. The dataset is limited to 11,055 tuples so as to reduce the impact of overfitting on the performance of the model. While preparing the dataset for the model, the 7 V's method has proven to provide the best results. i.e. the dataset should contain, Volume –the right number of tuples, Velocity – it should encompass the data from the present trend, Variety – it should contain all kinds of data attributes that supports or answers our problem, Variability – it should have such kind of data where it gives multiple meaning for different instances, Veracity – the data in the dataset should be accurate enough, Visualization – there should be valid relationship between the independent and dependent variables. This also helps in eliminating non-significant variables and to mine for patterns and Finally, The Value – it says about the usefulness of the dataset. With all these 7 factors fulfilled the dataset can be termed as complete to start the learning process. In our experiment the dataset of 11,055 tuples has acknowledge the seven factors. But this can change when other researchers start building the dataset. Yet another reason to stick with 11,055 tuples is that, random forest is just n decision trees to be described in brief. These trees are nonparametric machine

Table 3 A list of optimal values for the classifier as a result of GridSearchCV

S.No	Attributes	Values
1.	bootstrap	True
2.	ccp_alpha	0.0
3.	class_weight	None
4.	criterion	gini
5.	max_depth	None
6.	max_features	log2
7.	max_leaf_nodes	None
8.	max_samples	None
9.	min_impurity_decrease	0.0
10.	min_impurity_split	None
11.	min_samples_leaf	1
12.	min_samples_split	2
13.	min_weight_fraction_leaf	0.0
14.	n_estimators	100
15.	n_jobs	-1
16.	oob_score	False
17.	random_state	None
18.	verbose	0
19.	warm_start	False

learning algorithm. They are highly flexible and are subjected to overfitting the training data. The dataset with 11,055 records was finalized after testing with various sizes of datasets that ranged from 1000 to 30, 000 records. The dataset with too low or too high tuples or those datasets that does not follow the 7 V's has resulted in underfitting or overfitting or decline in accuracy. This dataset was finalized amongst others, based on the confusion matrix obtained for each after training the model on all the datasets.

Rule of extraction framework

Webpages possess a wide variety of properties. These properties can be used to distinguish a legitimate webpage with the phishing one (Mohammad et al. 2015). To list out the properties of the website, the rule of the Extraction Framework is used. This algorithm takes a URL as input and lists out 30 distinct features of a webpage that is used to determine its authenticity. These results are listed out and then fed to the Classifier Model for further processing. The 30 distinct features are listed below in Table 2.

Random forest classifier model

Random forest is an example of Ensemble learning. Random forest is a collection for n number of decision trees, where every decision tree produces different outputs for the same input. Here the majority of the outputs from n decision trees are considered as the output of the model. This model is trained on a dataset consisting of 11,055 tuples. To create a more effective model, we have used GridSearchCV to find the optimal parameters for the model. For the afore mentioned dataset, the best parameters produced by the function is listed in Table 3. The model is trained using K fold validation technique. Several experiments were performed on different number of K folds ranging from 2 to 12 on the dataset. The dataset set split into 5 K-folds produced the optimal result.

Based on the diverse dataset which is built and trained using the Random Forest Classification model, the trained model exhibits priority over the 30 features which highly contributes towards the identification of the website's category i.e. Legitimate or Phishing. This relative importance of these features as produced by the trained model is depicted in Fig. 5 more precisely.

Upon training the Random Forest Classification Model with the dataset along with the classifier parametric feature values mentioned in Table 3, it has produced more effective outcome. The Performance of the model is

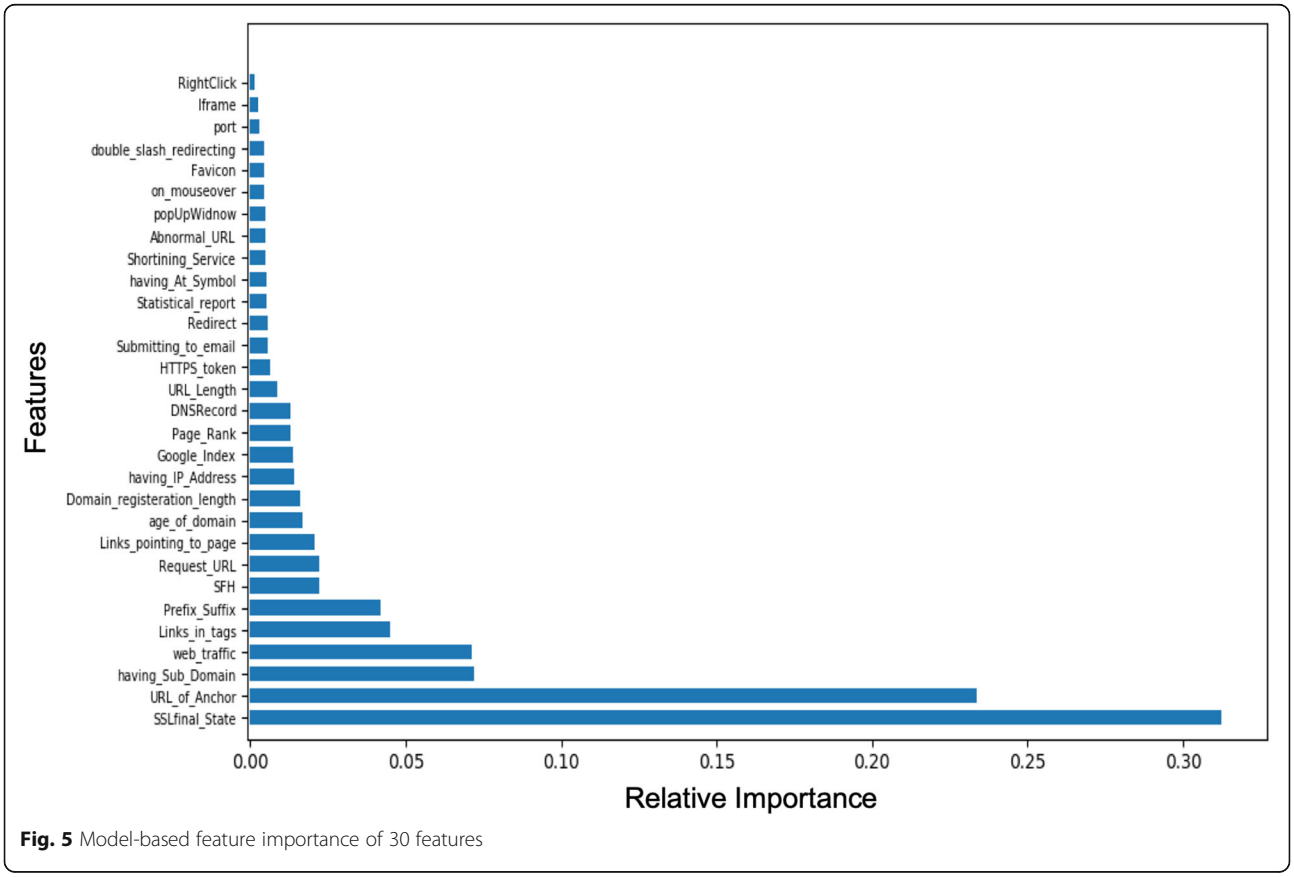

Fig. 5 Model-based feature importance of 30 features

analyzed through the confusion matrix produced over the test and predicted values. The more detailed information on the confusion matrix is represented in the Fig. 6 below for better and easy analysis.

Proposed EPDB system architecture workflow

The novelty of the redesigned browser in EPDB the implementation of the additional segment added into the existing architecture i.e. the Intelligent Engine. Firstly, the model is trained with the dataset using the k fold cross validation method along with the optimal parameters that we have obtained from gridsearchCV. Training the model requires time but predicting spans few seconds. Hence, the trained model is dumped into a pickle file. This pickle file is then loaded to predict a new instance. When a user requests for a website, the Browser engine is responsible for fetching the data from the internet. When this process is initiated, the URL is sent to the Intelligent Engine for verification in parallel. This engine uses Rule of Extraction Framework to determine all the 30 characteristics of the website. Then these 30 features are sent to the Random Forest Classifier algorithm for prediction. Since, the model is already trained and stored as pickle file. The file is loaded and decision is made on the authenticity of the URL received, and sent to the render engine. The render engine will receive the decision form the intelligent engine first, then it receives the data from the Browser Engine. If the decision received is Legitimate then the render engine will display the webpage requested from the user as shown in sec(i) in the Fig. 7. If the decision received was Phishing, then the render engine will freeze rendering the webpage and popup up an alert message saying 'The Website is Phish do you want to Continue…' as shown in sec(ii) in the Fig. 7. Then the user has two options, either to continue or to revert back to safety. This is not present in normal browsers as shown in sec(iv) in the Fig. 7.

The Main Advantages of the Proposed EPDB Architecture are:

- It is a real-time phishing website detection method with an accuracy of 99.36%.
- It has got 0% False Negative Rate.

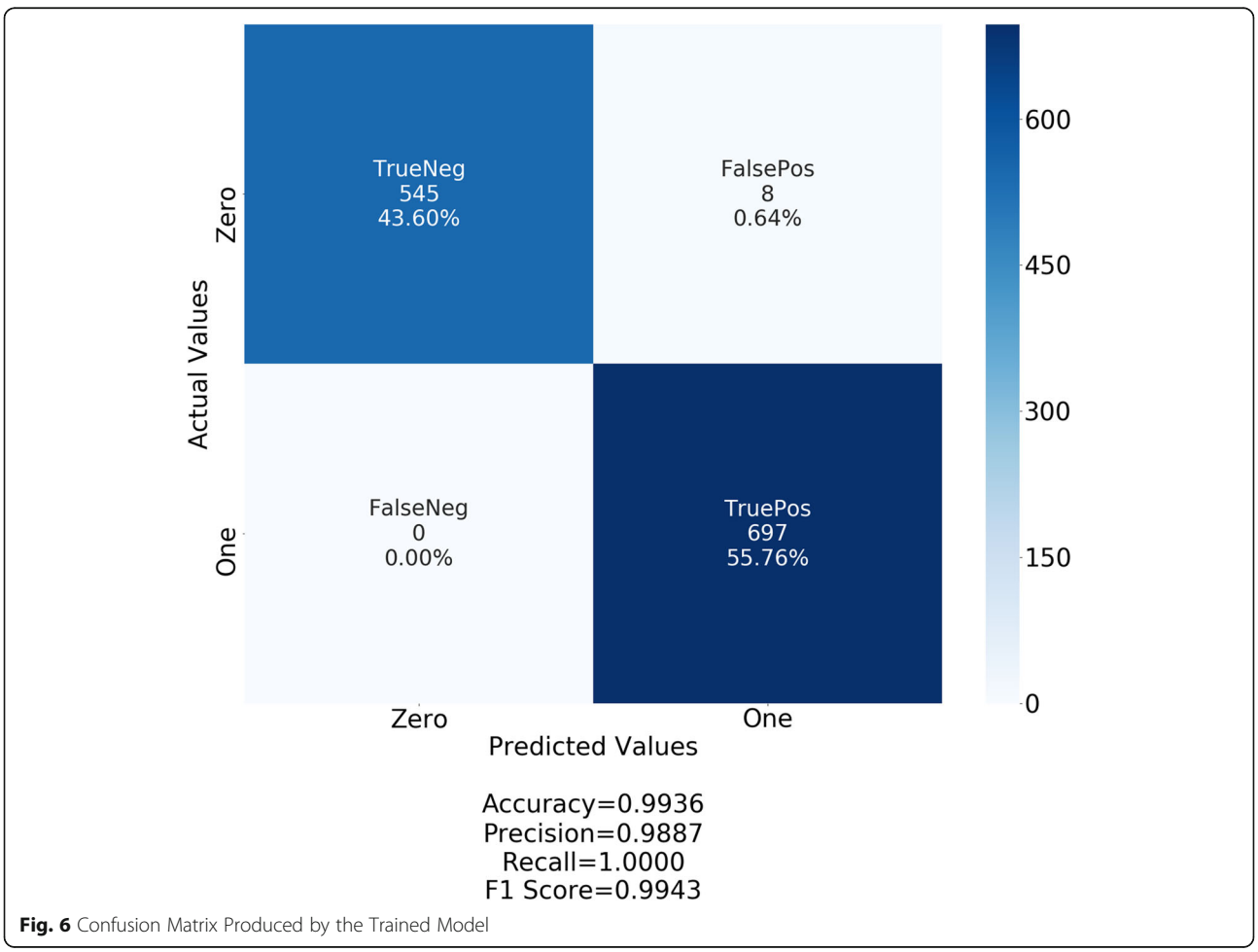

Fig. 6 Confusion Matrix Produced by the Trained Model

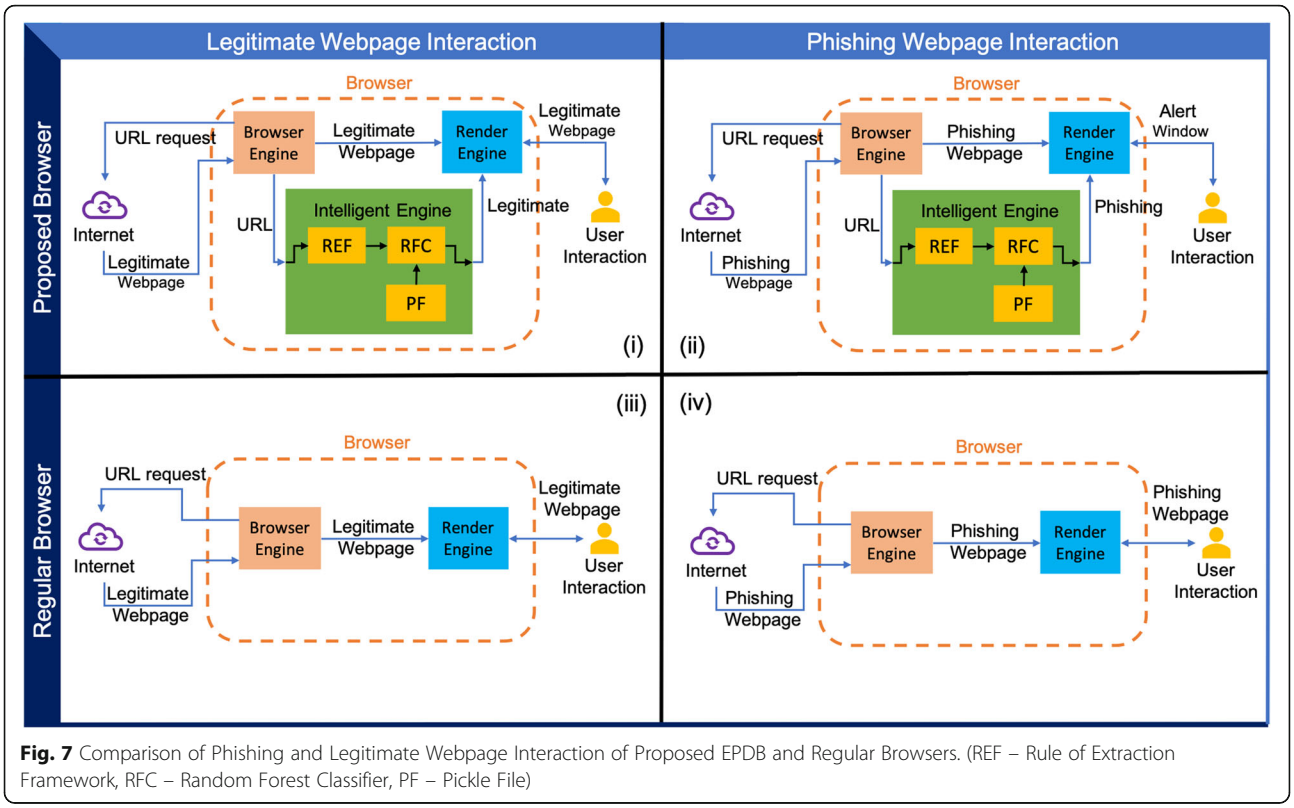

Fig. 7 Comparison of Phishing and Legitimate Webpage Interaction of Proposed EPDB and Regular Browsers. (REF – Rule of Extraction Framework, RFC – Random Forest Classifier, PF – Pickle File)

- Intelligent Engine and Browser Engine work in parallel. Hence, the browsing speed is not affected and there is no delay in the prediction.
- The website freezes if it is Phishing. Added advantage is that malicious JavaScript code if embedded as part of the website will never be initiated. This functionality is not a part of present browsers, i.e. current browsers like Chrome may show a website as Phishing but it still continues to execute the embedded malicious JavaScript code which is not possible with the proposed browser architecture.
- The pickle file is updated during browser updates with new pickle file obtained from training a new set of newly detected or collected datasets. Hence, the system will be capable of detecting new phishing websites adapting to the current trend.
- The Intelligent Engine uses very less memory and execution time because of the use of pickle file and prediction is not a complex process.

Performance analysis

A real-time performance analysis was done on the proposed EPDB method by developing a prototype. Later this was compared with the existing system. The following will have methodology on the proposed system and then with analysis results.

Methodology

To analyze the performance of the proposed EPDB system in real-time, a browser was developed from scratch with an intelligible interface. Then experiments were conducted to simulate the detection process. The experiment was done with Windows 10 on a dual-core Pentium processor of 2.1GHz with 2 GB of Memory and Kali Linux system. The Browser was developed using C# and Python and it was installed on Windows 10. A real-time phishing webpage was created using advanced tools on Kali Linux distribution. Then various attacks were launched using Kali Linux distro on Windows 10 PC running the newly developed Browser. The results were recorded.

Evaluation results

In the first experiment, we simulated a real-time attack on the browser that is been developed and the same attack was carried out on the world's most popular chrome browser. Surprisingly, the chrome browser did render the malicious webpage as it is, but the proposed EPDB as shown in Fig. 8, did recognize the webpage as phish and it popped up an Alert Message to the user about the attack.

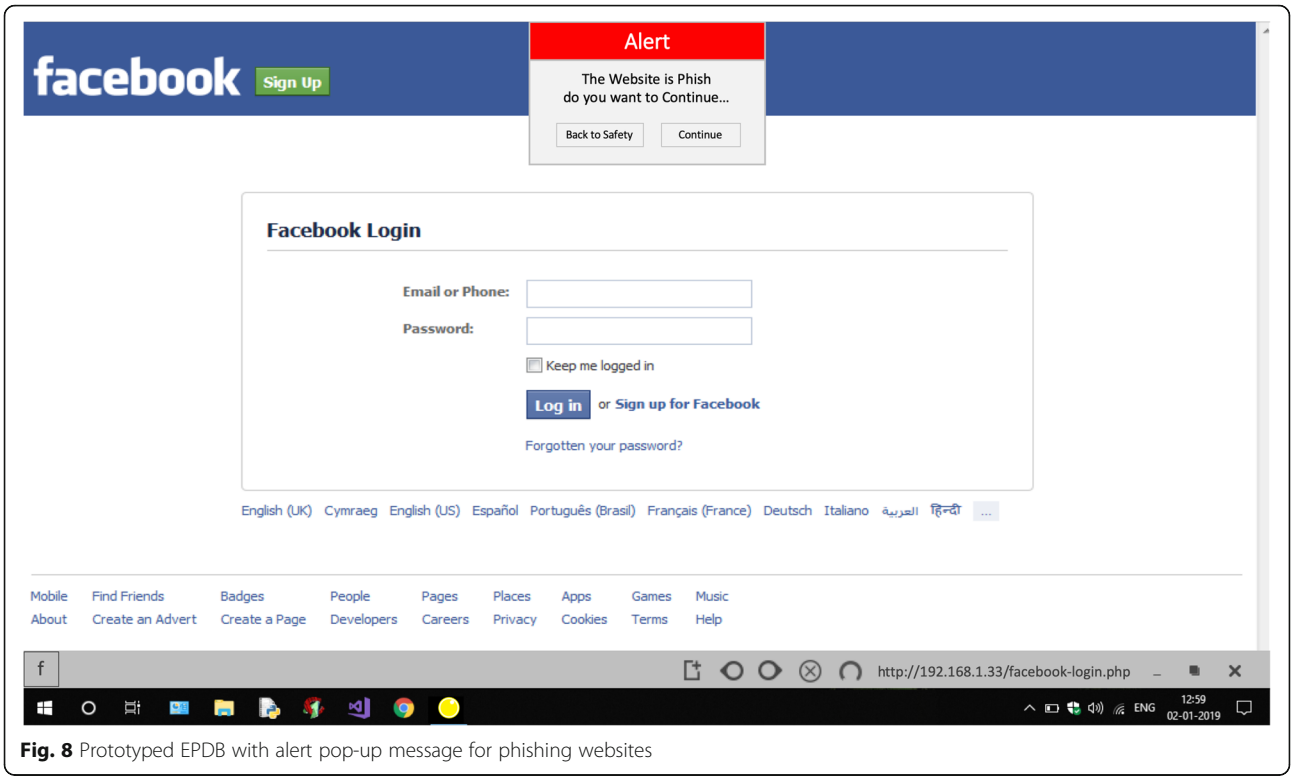

Fig. 8 Prototyped EPDB with alert pop-up message for phishing websites

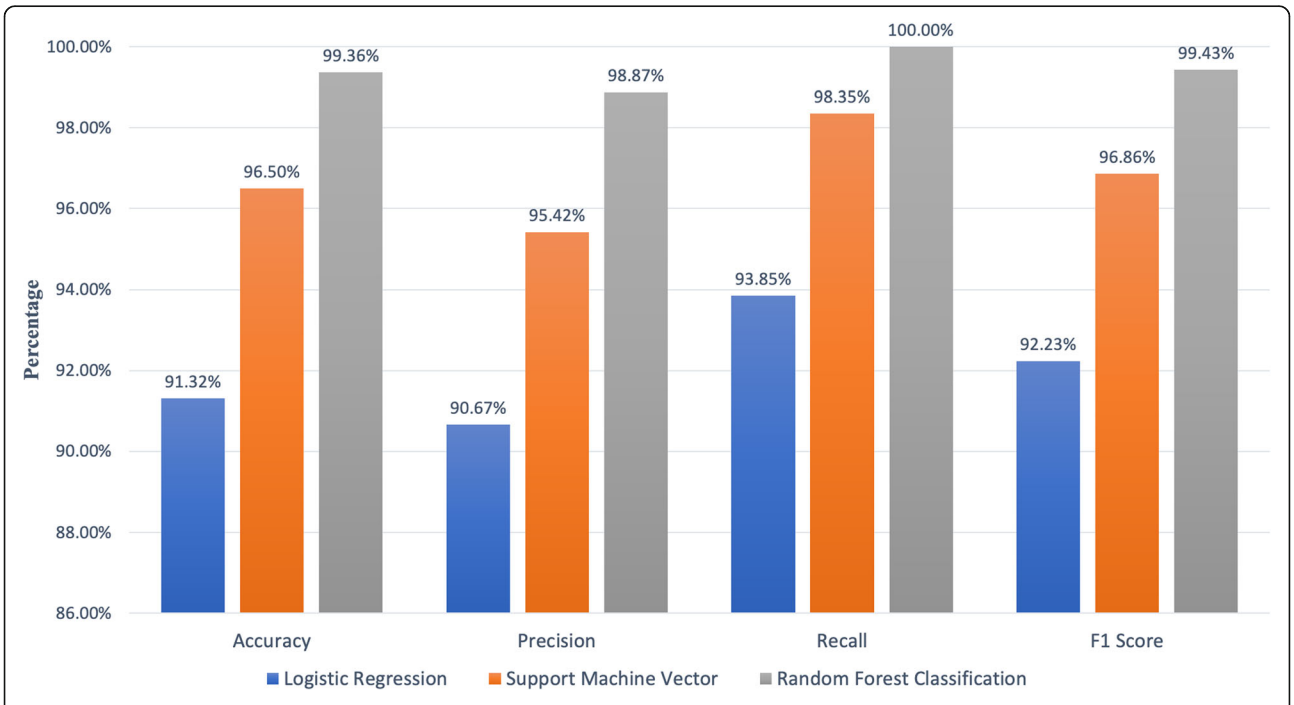

Fig. 9 Comparison of other models with the trained Random Forest Classifier

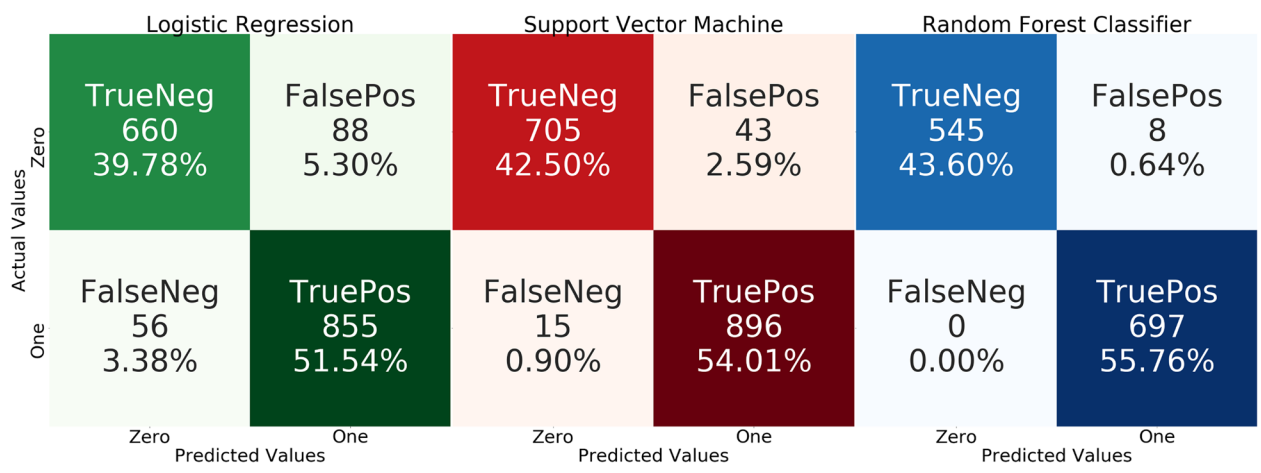

Fig. 10 Confusion Matrix of other models with the trained Random Forest Classifier

In the second experiment, we compared various other classification models i.e. Logistic Regression and Support Machine Vector with the random forest classification model. The Random Forest Classification model showed an accuracy of 99.36%, with a F1 Score of 99.43%, a detailed picture of comparison of Accuracy, precision, recall and F1 Score of the three models are visualized in Fig. 9. For a clear picture on the performance measure of the models, we have recorded the confusion matrix of all the three models and visualized as shown in Fig. 10.

In the third experiment, we tested the speed of our EPDB prototype with an existing chrome extension developed by other researchers as this being the most used method today. The prototyped EPDB model has taken around four-second on average to analyze the website and produce the result as show in Table 4. Where as the extension has taken around six-seconds on average depending on certain factors like system configuration, server response time, internet connectivity speed etc. The overhead on the EPDB protype model compared with the chrome extension is less by 33.3%. Through this experiment we could conclude that we could have a better speed while surfing through the web with faster response time. This factor of this prototype overcomes the disadvantages of server-side computations. i.e. Network connection speed, server-side computation overhead due to heavy traffic and server failures. Computing on the client-side makes this a stand-alone tool for filtering phishing websites.

Table 4 A comparison of performance overhead

Technique	Avg Response Time (Sec)
Embedded Phishing Detection Brower (EPDB - Proposed Prototype)	4
Phish Detector Chrome Extension	6

Final notes

In this paper, we proposed a secure web browser with all new browser architecture. This protects the user while surfing through the web by provides better security against phishing attacks in real time. The proposed prototype has performed well so far and has also got a new outlook which provides a wider view area for the webpage and UI. Apart from this, it is important to think about security while you are in the internet world, so the browser with an Intelligent engine protects the user from being hacked by phishing websites. The most interesting thing is that this engine will protect you from an attacker in real-time. The prototype provides a fast, reliable, and secure browsing experience for the users. As of now, the prototype lets the users gain the advantages of the security as well as basic features of the browser. In the future, the project can be helpful in various aspects of a normal person. For future scope and enhancement, the browser can be modeled to implement unsupervised learning. As of now, the browser uses a single trained model. In the future, when the user encounters a phishing website, the browser will register the website's URL in our server's. An updated version of the current dataset will be produced by collecting the phishing URLs from all users across the globe. After which a new model will be built using the new dataset and can be distributed to all the users via browser security update. This ensures the model to be trained with the most recent phishing websites.

Acknowledgements
'Not applicable'

Authors' contributions
The author(s) read and approved the final manuscript.

Funding
'Not applicable'

Availability of data and materials
'Not applicable'

Competing interests
The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Author details
[1]B.E in Computer Science and Engineering, PES College of Engineering, 4011, Vasuda Krupa, 3rd Cross, Shankar Nagar, Mandya, Karnataka 571401, India. [2]B.E in Computer Science and Engineering, PES College of Engineering, 1932, 1st Main Road, Halahalli Ext, Near Vinayaka Auto Stand, Mandya, Karnataka 571401, India. [3]B.E in Computer Science and Engineering, PES College of Engineering, 20/19, 5th cross , Shakthi Nagar, Near Shakthi Nagar Park, Mysore, Karnataka 570019, India. [4]Information Science and Engineering, PES College of Engineering, PES Engineering College Rd, PES College Campus, Mandya, Karnataka 571401, India.

Received: 20 February 2020 Accepted: 2 September 2020
Published online: 14 October 2020

References

2017 Internet Crime Report. (n.d.). Retrieved from https://pdf.ic3.gov/2017_IC3Report.pdf. Accessed 29 Aug 2020
Afroz S, Greenstadt R (2011) PhishZoo: detecting phishing websites by looking at them. In: 2011 IEEE fifth international conference on semantic computing, Palo Alto, CA, pp 368–375. https://doi.org/10.1109/ICSC.2011.52. Accessed 29 Aug 2020
APWG trends report q1 2019. (n.d.). Retrieved from https://docs.apwg.org/reports/apwg_trends_report_q1_2019.pdf. Accessed 29 Aug 2020.
APWG trends report q2 2019. (n.d.). Retrieved from https://docs.apwg.org/reports/apwg_trends_report_q2_2019.pdf. Accessed 29 Aug 2020
APWG trends report q3 2019 PRODUCTION. (n.d.). Retrieved from https://docs.apwg.org/reports/apwg_trends_report_q3_2019.pdf. Accessed 29 Aug 2020
APWG trends report q4 2019. (n.d.). Retrieved from https://docs.apwg.org/reports/apwg_trends_report_q4_2019.pdf. Accessed 29 Aug 2020
Armano G, Marchal S, Asokan N (2016) Real-time client-side phishing prevention add-on. In: 2016 IEEE 36th international conference on distributed computing systems (ICDCS), pp 777–778. https://doi.org/10.1109/icdcs.2016.44
Futai Z, Yuxiang G, Bei P, Li P, Linsen L (2016) Web phishing detection based on graph mining. In: 2016 2nd IEEE international conference on computer and communications (ICCC). https://doi.org/10.1109/compcomm.2016.7924867
Hawanna VR, Kulkarni VY, Rane RA (2016) A novel algorithm to detect phishing URLs. In: 2016 international conference on automatic control and dynamic optimization techniques (ICACDOT), pp 548–552. https://doi.org/10.1109/icacdot.2016.7877645
Hu J, Zhang X, Ji Y, Yan H, Ding L, Li J, Meng H (2016) Detecting phishing websites based on the study of the financial industry webserver logs. In: 2016 3rd international conference on information science and control engineering (ICISCE), pp 325–328. https://doi.org/10.1109/icisce.2016.79
Jain AK, Gupta BB (2017) Phishing detection: analysis of visual similarity based approaches. Secur Commun Netw 2017:1–20. https://doi.org/10.1155/2017/5421046
Join the fight against phishing. (n.d.). Retrieved from https://www.phishtank.com/. Accessed 29 Aug 2020
Ma J, Saul LK, Savage S, Voelker GM (2009) Beyond blacklists. In: Proceedings of the 15th ACM SIGKDD international conference on knowledge discovery and data mining - KDD 09. https://doi.org/10.1145/1557019.1557153
Mao J, Tian W, Li P, Wei T, Liang Z (2017) Phishing-alarm: robust and efficient phishing detection via page component similarity. IEEE Access 5:17020–17030. https://doi.org/10.1109/access.2017.2743528
Marchal S, Armano G, Grondahl T, Saari K, Singh N, Asokan N (2017) Off-the-hook: an efficient and usable client-side phishing prevention application. IEEE Trans Comput 66(10):1717–1733. https://doi.org/10.1109/tc.2017.2703808
Mei C, Leng C, Dayang H, Abang I, Nah S (2016) Feature-based phishing detection technique. J Theor Appl Inf Technol:101–106 Retrieved from https://ir.unimas.my/id/eprint/13943/
Mohammad, R. M., Fadi, T., & Lee, M. C. (2015). Phishing websites features. Retrieved from http://eprints.hud.ac.uk/id/eprint/24330/6/MohammadPhishing14July2015.pdf
Pandey M, Ravi V (2013) Text and data mining to detect phishing websites and spam emails. In: Panigrahi BK, Suganthan PN, Das S, Dash SS (eds) Swarm, evolutionary, and memetic computing. SEMCCO 2013. Lecture notes in computer science, vol 8298. Springer, Cham. https://doi.org/10.1007/978-3-319-03756-1_50
Phishing scams and spoof emails at MillerSmiles.co.uk. (n.d.). Retrieved from http://www.millersmiles.co.uk/. Accessed 29 Aug 2020
Phishing Trends & Intelligence Report 2018. (n.d.). Retrieved from https://info.phishlabs.com/hubfs/2018PTIReport/PhishLabsTrendReport_2018-digital.pdf. Accessed 29 Aug 2020
Roy S, Patra A, Sau S, Mandal K, Kunar S (2013) An efficient spam filtering techniques for email account. Am J Eng Res 02(10):63–67 Retrieved from http://www.ajer.org/papers/v2(10)/F02106373.pdf
Sahingoz OK, Buber E, Demir O, Diri B (2019) Machine learning based phishing detection from URLs. Expert Syst Appl 117:345–357. https://doi.org/10.1016/j.eswa.2018.09.029
Williams N, Li S (2017) Simulating human detection of phishing websites: an investigation into the applicability of the ACT-R cognitive behaviour architecture model. In: 2017 3rd IEEE international conference on cybernetics (CYBCONF). https://doi.org/10.1109/cybconf.2017.7985810
Wu C, Kuo C, Yang C (2019) A phishing detection system based on machine learning. In: 2019 international conference on intelligent computing and its emerging applications (ICEA), Tainan, Taiwan, pp 28–32. https://doi.org/10.1109/ICEA.2019.8858325

Publisher's Note