Browser Extension based Hybrid Anti-Phishing Framework using Feature Selection

Swati Maurya[1]
Guru Gobind Singh Indraprastha University
New Delhi, India

Harpreet Singh Saini[2]
New Delhi, India

Anurag Jain[3]
Guru Gobind Singh Indraprastha University
New Delhi, India

***Abstract*—Phishing is one of the socially engineered cybersecurity attacks where the attacker impersonates a genuine and legitimate website source and sends emails with the intention of stealing sensitive personal information. The phishing websites' URLs are usually spread through emails by luring the users to click on them or by embedding the link to fake website replicating any genuine e-commerce website inside the invoice or other documents. The phishing problem is very wide and no single solution exists to mitigate all the vulnerabilities properly. Thus, multiple techniques are often combined and implemented to mitigate specific attacks. The primary objective of this paper is to propose an efficient and effective anti-phishing solution that can be implemented at the client-side in the form of a browser extension and should be capable to handle real-time scenarios and zero-day attacks. The proposed approach works efficiently for any phishing link carrier mode as the execution on clicking on any link or manually entering URL in the browser doesn't proceed unless the proposed framework approves that the website associated with that URL is genuine. Also, the proposed framework is capable to handle DNS cache poisoning attacks even if the system's DNS cache is somehow infected. This paper first presents a comprehensive review that broadly discusses the phishing life cycle and available anti-phishing countermeasures. The proposed framework considers the pros and cons of existing methodologies and presents a robust solution by combining the best features to ensure that a fast and accurate response is achieved. The effectiveness of the approach is tested in a real-time dataset consisting of live phishing and legitimate website URLs and the framework is found to be 98.1% accurate in identifying websites correctly in very less time.**

***Keywords*—Anti-phishing; browser extension; machine learning; feature selection**

I. Introduction

Phishing attack is the cyber threat that has been prevailing on the internet for almost four decades despite having very exhaustive research in this area and numerous anti-phishing solutions and prevention techniques being available. Phishing is one of the prominent cyber-attack that involves sending emails with the intention of obtaining sensitive personal information by pretending to have been sent by a trustworthy or genuine sender. It is a social engineering attack that exploits the weakness found in the user's systems. It is performed over different channels ranging from sending of fake email by the attacker to fabricated fake websites, social networks or even cloud services [1]. Phishing is an automated identity theft activity, which takes the advantage of human nature and loopholes in technology to lure general users to click on fabricated fraudulent hyperlinks that lead to serious consequences for the victim by breaching the data and misusing the credentials for malicious activities. Despite having high-security implementation in internet browsers, phishing attacks succeed in breaching the security approaches.

– Phishing attack life cycle: Phishing campaign is a four-step process that starts with planning and setting up the prelims for the attack in which the target group is chosen and attack techniques are analyzed and finalized. After the attack, results gathered from the attack are processed to gather useful information out of the data collected. The steps involved in the phishing process are shown in Fig. 1.

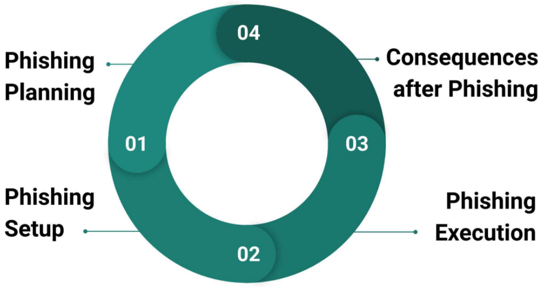

Fig. 1. Steps involved in Phishing attack life cycle

– Phishing Planning: The first step in the Phishing process is planning the attack. The adversaries choose communication media and attack targets.

– Phishing setup: In the second step, the attacking techniques are finalized. The methods for sending the email containing suspicious link and creating the fraud/phished webpage is performed in this step. This step is also known as Phishing preparation.

– Phishing Execution: The important step is performing the attack. It includes Propagation and Penetration steps. The attack material is propagated to the targets. As the target opens the attack material, either the user is prompt to enter personal information supposed to be stolen or a malware or malicious application is downloaded into the target system.

– Consequences after phishing: The phishers execute the information received through the phishing campaign. Attack results, i.e., the sensitive credentials that attack victims entered, are exploited. They misuse them to make illegal fund transactions and commit fraud.

A recent report by Anti-Phishing Working Group (APWG) shows that the number of unique phishing attack reports

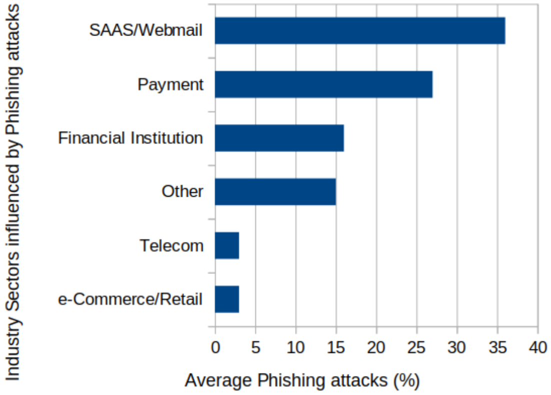

Fig. 2. Most targeted industry sectors for Phishing attacks [2]

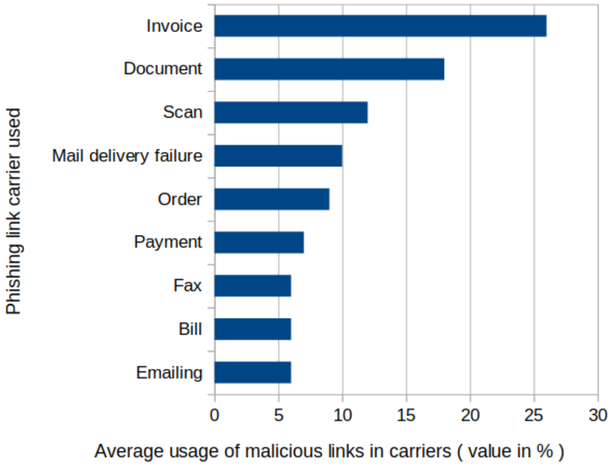

Fig. 3. Fabricated malicious link carrier tactic used for Phishing [6]

submitted during the 1st quarter of 2019 was 1,80,786 nearly 30.7% more than 1,38,328 reported in 4th Quarter 2018 [2]. According to ProofPoint's State of the Phish 2019 Report, 83% of global security professionals encountered phishing attempts [3]. Most affected industry sectors by phishing attacks are represented in Fig. 2. 255065 unique phishing attacks were reported worldwide till 2016. This represents an increase of over 10% from the 230280 attacks that were identified in 2015 due to an increase in 'Domain Name Use' attacks [4]. According to Symantec's 2018 Internet Security Threat Report [5], Sending malicious attachment with the email having fake invoice is the most popular tactic for luring the common public to open the message and click the fabricated link. The percentage of each tactic used for sending fabricated links in Phishing email is shown in Fig. 3.

Most of the anti-phishing solutions in literature claim high accuracy as 98% for phishing detection but most of these measures fail to handle real-time zero-day attacks. There is a huge gap between the high accuracy that has been reported in articles but when it comes to real-time scenario implementation, most of the existing solutions have very low effectiveness. Thus an effective and efficient anti-phishing framework needs to be designed that should provide a fast and accurate response and should be easy to implement. Client-side solutions in the form of browser extensions are easy to install and use. The computational cost associated with browser extensions should be as minimum as possible otherwise it will drastically affect the response time.

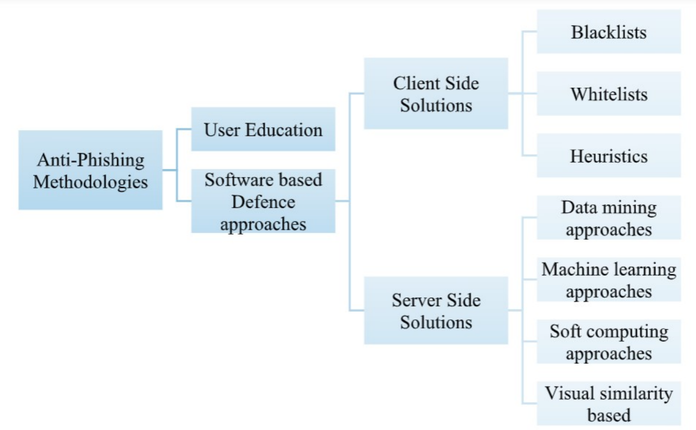

Fig. 4. Social and Software-based Anti-Phishing methodologies

The proposed framework is a hybrid client-side solution that takes the pros of existing anti-phishing methodologies and tries to minimize their associated cons. The paper is structured as follows: Section 2 discusses the literature survey on existing anti-phishing methodologies. The motivation behind the proposed framework is discussed in detail in Section 3 and Section 4 provides system architecture of the proposed framework and its effectiveness in real-time data set that consists of live phishing and legitimate websites. Section 5 represents the results and analyze the observations of the experiments. And finally in Section 6, the discussion about the outcomes of the proposed framework is done their limitations and other issues are also discussed.

II. Literature Survey on Existing Anti-Phishing Methodologies

Mitigation of phishing attacks consists of Detection, Prevention and Correction approaches. The phishing problem is very wide and no single solution exists to mitigate all the vulnerabilities properly. Thus, multiple techniques are often combined and implemented to mitigate specific attacks. Automatically detecting phishing web pages has attracted much attention from security and software providers, financial institutions, to academic researchers. Several approaches and comprehensive strategies have been suggested to tackle phishing. Anti-phishing methodologies can be grouped broadly into the categories shown in Fig. 4:

- User Education

Phishing attacks usually target the users who are not much aware of the defense mechanisms for phishing. Since phishing attacks attempt to take advantage of the inexperienced users, an obvious solution is educating the users, which would, in turn, reduce their susceptibility to falling victims of phishing attacks [7]. Several user training approaches have been proposed throughout the past years. The human factor is broad. Relying solely on educating users about the phishing attacks and prevention mechanisms does not ensure protection from Phishing attacks.

- Software-based Defence Approaches

Software-based defense approaches can be classified according to the place where they are implemented. If the defense mechanism needs to be implemented at the client's system either in the form of a browser plug-in or toolbar or any desktop application, it is known as a Client-side solution. Server-side techniques depend upon the classification performed on the server-side.

1. Client-Side Solutions

These include user profile filters and browser-based toolbars. These tools also depend on blacklisting and whitelisting techniques where a list of detected phishing or legitimate websites is downloaded with updates at standard intervals.

– Profile Matching (Blacklists and Whitelists)

Blacklists contain suspicious URLs that might redirect to the Phishing webpage, IP addresses, and keywords. These approaches rely on previously detected attacks database and need to be updated frequently. Blacklists are generally implemented as toolbars or web browser extensions. Common examples are Mozilla Firefox's extensions, Google safe browsing API [8], PhishTank [9] and many more. Whitelists act as a filter that contains trusted and reliable URLs and IP addresses which are marked safe and legitimate websites. The rest of the content is treated suspicious and undergoes scrutiny. Blacklists are found to be less effective as compared to whitelists as they cannot deal with 'Zero-day attacks'. Whitelists suffer from a large number of false positives as the lists contain only a limited number of legitimate websites and everything else is considered to be a phishing website.

– Heuristic or rule-based methods

Heuristic techniques are directed to solve the phishing problem through a practical method that utilizes the characteristics present in a phishing attack. The heuristics are identified from general phishing attacks and used for future detection and hence are efficient for zero-hour phishing attacks.

2. Server-side solutions

Server-side defense mechanisms are based on content filtering approaches and are appropriate to fight zero-day attacks. The filters are based on the following techniques:

– Machine learning approaches

The anti-phishing approach where the data is input to complex algorithms that learn the patterns in the given data by mining the insights and takes decision is known as the Machine learning approach. It applies to structured as well as unstructured data sets. It is used for developing techniques for knowledge extraction from datasets based on Artificial Intelligence and Statistics [10]. The representation model formed with the help of this knowledge is used for predictions for new data. Mostly used machine learning classifiers are naive Bayes classifiers, support vector machines (SVM), k-nearest neighbors, Decision Tree, Random Forest, Boosting and TF-DIF.

- Decision Trees (DTs): Decision-tree learning is a symbolic induction method that produces syntactically simple, easily interpreted rules [11]. In DTs, the knowledge extracted from the given data is organized in a recursive hierarchical structure represented with the help of nodes and branches [12]. DTs aim to maximize the correct classification of all training data. The pruning technique is applied to the trained tree to avoid the problem of overfitting [13]. DTs provide the advantage of verifying the attributes that determined the final classification [14].

- Random Forests: A random tree is a tree that is formed by a stochastic process. Random forests are combinations of random tree predictors [15]. They are an ensemble learning method for classification and regression that overcome the problem of overfitting in decision trees.

- k-Nearest neighbor: k-NN is the simplest machine learning technique. It is an instance-based learning, also known as lazy learning, that stores all training data and classifies a new data point according to the class of the majority of its k-nearest neighbors in the given feature space [14]. For finding the nearest neighbors for each data, different measures are used to calculate the distance between pairs of data nodes. Euclidean distance is the most commonly applied measure.

- Naive Bayes: NB are the probabilistic classifiers based on Bayes theorem. The classifier learns the conditional probability of each attribute value from the training data given the classification of each instance. For the classification of an unknown instance, Bayes' theorem is applied. Naïve Bayes training is usually performed by using maximum likelihood algorithms.

- Support Vector Machines: SVM are linear classifiers based on Statistical Learning Theory [16]. SVMs perform structural risk minimization, for improving the complexity of the classifier. SVM constructs a hyperplane that optimally separates the data into two categories in higher-dimensional space.

– Data Mining Techniques

The techniques that come under this category consider phishing to be a classification or clustering problem and algorithms based on data mining with the help of machine learning techniques are applied to them.

– Soft Computing Techniques

Soft Computing is the fusion of methodologies that are applied to real-world scenarios to find the optimum solution to the problems that are not easily modeled mathematically. In this approach, knowledge discovery is used to simplify the evolution process. The evolving clustering method for classification and approaches based on Fuzzy neural networks is used to develop a model for phishing detection which performs the classification using some features to classify phishing and legitimate emails.

Various anti-phishing research works that are based on profile matching, heuristics approaches, machine learning algorithms, and soft computing techniques are discussed below:

Kirda and Kruegel (2005) presented "Anti-Phish" [17], a Mozilla browser extensions that captures and stores sensitive information using Paul Tero's Javascript DES implementation with the context and domain of the website where it was submitted. Whenever a user enters any credentials on a website, Anti-Phish compares the records in the watch list and if the same values are being entered, the domain is examined thoroughly. If the domain is different from the domain in trusted records, it is assumed to be a phishing website and an alert is given to the user. This approach ensures that any sensitive information is not forwarded to any suspicious website. The approach is efficient to monitor javascript hooks embedded in HTML webpages to capture data on keystrokes and send this information to the malicious server in the background even without the user submitting the details.

Prakash et al. (2010) proposed a system that combines two components: URL prediction and Approximate URL matching for addressing the exact match of URLs limitation in case of blacklist approaches [18]. New malicious URLs are predicted from the existing entries in the blacklist and are tested whether they are malicious through ‘DNS queries’ and ‘Content matching’. For generating new URLs, five heuristics, namely, ‘Replace Top-level domains (TLD)’, ‘IP address equivalence’, ‘Directory Structure similarity’, ‘Query string substitution’ and ‘Brand name equivalence’ s utilized. The new URL, after creation, is subjected to a validation process that uses the DNS Lookup mechanism to filter out URLs that do not exist or are marked as legitimate. For the approximate matching process, the input URL is broken into 4 entities- IP address, hostname, directory structure, and brand name. These entities are matched with corresponding fragments of the blacklist entries and based on the matching score; URLs are marked as Phishing or Legitimate. This technique is capable of fast detection but suffers the drawback of too high false negatives, i.e., 5%.

Belabed et al. (2012) proposed an extension for web browsers that combines a personalized whitelist and SVM classifier [19]. The whitelist is implemented in the form of an XML file which consists of user’s login pages’ URLs and a set of keywords that is composed of domain names of the page’s URL and terms from DOM tree for the website. - “Bag of words” model [20] is used for constructing keywords’ frequency vector and Cosine distance [21] to find the similarity between the visited webpage and the webpages available in the whitelist repository. A similarity check is processed as explained in Fig. 5. A feature vector that represents the webpage utilises 8 features according to its URL and content (URL with IP address, Special characters in the URL, presence of SSL certificate, whether the identity of the webpage conforms to its URL, search engine ranking, nil anchors, frequency of links, action complies with the page identity) is processed by the classifier. This approach can detect 98% phishing webpages accurately. The major drawback of this approach is the high false-positive rate, i.e, 3.5%.

Islam and Abawajy (2013) proposed a “Multi-tier classification model” for filtering phishing emails that combines multiple classification algorithms to reduce false positives and increase the overall efficiency [22]. Based on the weighting of message content and message header, the features from an email are extracted and prioritized according to their ranks. Different combinations of classifier algorithms in a multi-tier classification process are tested and the impact of rescheduling is examined. 21 features are used for classifying emails and Spamassassin public corpus, 2006 and Phishing Corpus homepage 2006 dataset is used. For 1st tier- SVM, 2nd tier- AdaBoost and for 3rd tier- Naive Bayes Classification algorithms are used. This approach attained the accuracy up to 97%, at most 2% FP and at most 9% FN.

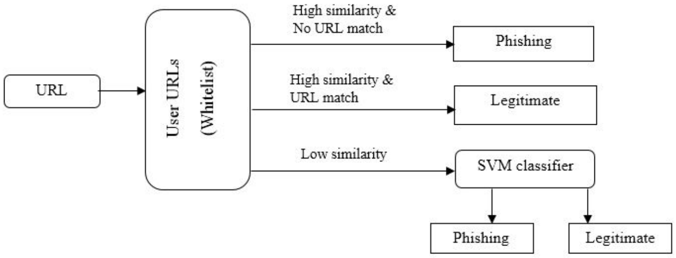

Fig. 5. Classification based on similarity with webpages in whitelist in [19]

Akinyelu and Adewumi (2014) proposed a content-based phishing detection approach that works for email filtering [23]. A set of 15 features selected from the literature is used for forming a vector representation by extracting values from a dataset of 2000 emails and then the Random Forest classifier is tested and trained using 10-fold cross-validation. In this classification, before decision tree construction, information gain for all the features is calculated and the features that are found to have the best 8 information gain construct the decision tree. Prediction of emails is done by mode vote of all the trees. This algorithm assured the accuracy of 99.7% and true positives are quite low, i.e., 0.06% whereas false negatives are too high (2.5%).

Gowtham and Krishnamurthi (2014) proposed a hybrid approach that combines preliminary filters that eliminate webpages not containing login forms and 15 heuristics based on structural and behavioral properties of webpage [24]. The system architecture consists of: Module 1- Preapproved site identifier: A self-constructing private whitelist is used for prevention from pharming attacks. This module contains legitimate websites checked with remote DNS lookups. When the user accesses any website, this module checks the URL and IP address of the website and declares it legitimate if found in the whitelist or else forwarded to the next module. Module 2- Login Form Finder: In this module, 38 login keywords and a search keyword are used to verify whether a webpage has a login form or not. The webpages without login form are marked safe and stopped from entering the next step. Webpage Feature generator: The webpage’s identity is extracted from its hyperlinks and contents and the heuristics clustered into six groups check the phishing characteristics of the webpage. A 15- dimensional vector is formed by domain descriptor. SVM classifier: The feature vector generated is inputted for classification. The rules developed during the training period are used for predicting the class. The class label of the classification of new test cases is the output based on the patterns found in training data. This approach is tested over a dataset of 2464 live websites from which 700 sites are legitimate and 1764 are phishing sites. TPR of 99.6% and a reduced FPR of 0.42% are achieved.

Rao and Ali (2015) developed a desktop application based on 5 modules that input the URL of the website and declare the status of the website as a legitimate or phishing [25]. This approach processes only the websites containing the login form. Module 1- Whitelist: Comparison of the domain of URL with that of the genuine websites stored in the database is done and the legitimate status is set for the sites that match in the list. The sites that do not match are passed to the next level. Not the entire webpage but its DOM is passed to the next module. Module 2- Zero links in the body portion of HTML indicate that the body section of the webpage does not have any link embedded and text is appearing in the form of images. This heuristic if marked zero represents a phishing site and stops the process or else proceeds to the next step. Module 3- Footer links pointing to NULL: In any legitimate site, the footer link never points to null. This heuristic is used to classify the websites as phishing those have the anchor tag in the footer section pointing to null. Module 4- Copyright and Title content is extracted and compared with the sites in whitelist. A match found represents a phishing website. Module 5- Website Identity: This reveals the original targeted website that the phished sites trying to pretend. PhishShield application is based on Jsoup API and Firebug. Whitelist used is based on PhishTank's target list [9]. For evaluation of the approach, 1600 phishing sites and 250 legitimate sites are tested. The major limitation of this approach is its inability to detect phishing websites that do not have any resemblance with or try to imitate any of the legitimate websites and can successfully bypass all the filters used in this approach.

Jain and Gupta (2016) proposed an anti-phishing solution in the form of a "browser plug-in" that matches the current domain with the legitimate domains defined in the whitelist [26]. The proposed approach works through two modules: URL and DNS matching module matches the domain of the webpage and IP address from the whitelist. The phishing identification module uses 'hyperlink features' which are extracted from the DOM object properties of a webpage using Jsoup [27] to check the legitimacy of the website. Phishing detection algorithm is applied after this which takes decision based on three parameters of the hyperlinks whether the webpages contain any hyperlink, whether the webpage contains null pointer and the number of links that are pointing to its own domain and outside domain. The system can detect DNS poisoning, embedded objects and zero-hour attacks with an accuracy of 89.38%, TPR=86.07%, and FNR=1.48%.

Yang et al. (2017) analyzed and evaluated the performance and effectiveness of the C4.5 algorithm for detecting phishing websites [28]. They proposed a decision tree using C4.5 for classification of websites into phishing and legitimate. The dataset used (PWD) is obtained from the UCI machine learning repository that consisted of 11055 websites (4898 phishing websites + 6157 legitimate websites). The technique used 30 features of Phishing website dataset which are partitioned into four classes: Address bar based features, abnormal based features, HTML and JavaScript-based features, and domain-based features. To study the effect of dimension reduction of features on the performance of the decision tree, a dataset PW2 is generated from PWD using the selection attributes method. PW2 contains 9 features. After applying C4.5 on PWD and PWD2, it has been observed that PWD2 is better than PWD in terms of complexity and computational cost. However, the results show that reducing the number of features for classification decreases the accuracy slightly. Hence, it can be concluded that with more features, classification is better.

Li et al. (2019) collected real-time websites' data to form two datasets, namely, 50K-PD and 50K-IPD [29] Their approach is to extract features by using Word2Vec model to extract HTML string embedding by learning distributed representations of words and combine with features extracted from URL. In the next step, Gradient Boosting Decision Tree, LightGBM and XGBoost machine learning models are combined to form a stacking model. The predictions of training set obtained by using these 3 models are combined with the original feature set to form a new training set which is fed to the next layer of this stacking model. In the end, the GBDT model is used for the final decision. The claimed accuracy is 96.45%.

Ding et al. (2019) used Search engine based detection technology combined with heuristics to classify websites into legitimate and phishing [30]. The approach enters the title tag of the webpage as a search keyword into 'Baidu' search engine and marks it as legitimate if the website is within the top ten results from searching and skips the next steps. If the result is not in the top 10 results, URL heuristics of the webpage are matched with the defined rules and classify the webpage as phishing if it matches. If this step fails to mark the webpage as phishing, the logistic regression classifier is fed with "URL's DNS, Whois, similarity with phishing vocabulary, lexical feature and HTML" and the classifier results with the final prediction as phishing or legitimate website. The approach has been tested with datasets from Phishtank, Yahoo, URLB, and DMOZ and claims to have an accuracy of classification as 98.9%.

III. Motivation for the Proposed Framework

Various anti-phishing solutions proposed by different authors have been given in the previous section. However, no single solution is a "full proof" solution for combating phishing attack. Limitations of the existing anti-phishing solutions emphasize the need for innovative solutions. Some organizations provide guidelines and cybersecurity policies to be used by the common user as best practices for prevention from online fraud and phishing attacks [31]. The effectiveness of an anti-phishing solution depends on its capability to recognize a phishing website or email within an acceptable time period. As seen in the literature, numerous anti-phishing solutions are available, but most of them are unable to take highly accurate and precise decisions. In most of the techniques, a rise in false positives has been observed, i.e., classifying legitimate websites as a phishing website.

Every anti-phishing approach has its own associated pros and cons:

- The blacklist and whitelist approaches have been observed to have too high false positive and false negative rates. The drawback of the blacklist approach is that the blacklist cannot be updated frequently and does not provide 100% coverage of all phishing websites. Hence, they alone are not effective for zero-day phishing attacks.

- The approaches based on heuristics that use several website features for identifying the type of website are much more successful for phishing detection as compared to list matching approaches. They are quite effective in detecting fake websites in real-time but some of them have very high FP rates.

- The machine learning and data mining approaches give the best results in phishing detection. But the selection of appropriate classifier is a challenging problem. There are pros and cons associated with every classifier. SVM classifies the webpages with high accuracy but this approach is very time-consuming and is often best used for small datasets. Naïve Bayes classifiers are easily implemented but require the features to be mutually independent. Machine learning techniques require systems to have high computation power to be implemented in real-time but are the most effective ones. It can be concluded from the above sections that Support Vector Machine (SVM), Random Forest and Logistic regression-based classifiers are the most commonly used classifiers in the literature that has been covered for this study.

Most of the anti-phishing solutions in literature claim high accuracy as 98% for phishing detection but most of these measures fail to handle real-time zero-day attacks. There is a huge gap between the high accuracy that has been reported in articles but when it comes to real-time scenario implementation, most of the existing solutions have very low effectiveness. A few major reasons for low-effectiveness of most of the existing solutions are:

1. The design and ideology behind the solution is influenced by high detection accuracy obtained by training the classifiers with dataset having limited features or spatial correlation.

2. The model used for testing the evaluation of anti-phishing solutions are not capable of representing the real-time scenarios effectively.

The major requirements for an effective anti-phishing solution is to have these characteristics:

- The detection performance should be evaluated in real-time scenarios after considering all the use cases and deployment cases.

- The temporal resilience of the dataset is a must.

- The evaluation or assessment methodology should be fast enough to provide efficient results in fractions of seconds.

As discussed in Section 3, anti-phishing solutions can be provided either on the client-side or server-side. Most of the phishing victims are tricked using emails through the webpage they open via link provided in the email. An effective implementation is considered to be the one that gives a prompt decision in the form of an active warning to the user if he is trying to access any webpage that is a phishing webpage. Client-side anti-phishing solutions seem to help in a better way by providing results immediately. The proposed solution is a client-side solution in the form of “browser extension” which gives user prompt alert if the requested webpage is classified as “Phishing” by the detection model.

Few considerations while designing a client-side solution:

Since the main motivation behind adopting a client-side solution is to maintain user privacy and super-fast decisions, a few major points should be considered side by side. The ideal solution should not demand high computational power and since it is going to be integrated with a web browser, it should not degrade the performance of the client-side. The client should be kept engaged or given proper messages in case it is taking any delay even though for seconds as user interaction is much needed. The client-side based solution should always maintain the confidentiality of the detection model.

IV. System Architecture of the Proposed Framework

The basic architecture of the proposed browser extension is shown in Fig. 6. As shown in the figure, this framework is divided into three layers:

1. Whitelist check

The top-most layer contains a list of trusted websites that have been marked as safe by reputed agencies. For now, the records have been taken from Alexa Top Sites [32]. When a URL is requested in the browser either by clicking through e-mail or by manually entering it in the address bar, the browser extension checks the whitelist whether the domain of the requested webpage matches in the records. If the domain matches, IP address is also checked and if it also matches, the webpage is declared “Legitimate”, otherwise, “Phishing”. If the domain doesn’t match, the request is passed to the next layer. By matching domain with the IP address, the proposed solution works against the “DNS cache poisoning” attacks and identifies them.

2. Blacklisting websites

This layer acts as a “filter” which stops the webpages that are phishing webpages. For providing highly accurate results, this solution relies on experts from “PhishTank” for the webpages that they have marked as phishing after thorough examination. This process saves time that would have been consumed in evaluating the same webpages at its own end and hence gives the result very fast.

This layer queries API provided by “PhishTank”. There are two types of responses from the Phishtank API. Response string contains “Valid” if the webpage has been declared as “phishing” by their experts. Response is “Unknown” if the website is under evaluation in their system or it is not found in their record. Thus, if ”valid” is returned, the process stops here as the webpage is Phishing webpage and the user is given an appropriate alert stating that its a Phishing website. In the case of “unknown”, the request is passed to the next layer of the extension.

3. Feature extraction and classifiers

This layer consists of three phases: ‘Form evaluation’, ‘Hyperlinks extraction and analysis’ and ‘Stack of classifiers’. For extracting features from the webpage, the rules have been used to define condition or range for features as described in Table I.

Phase 1: Form evaluation

The webpage is scanned in this phase to check the presence of any form. If the form is not present, the webpage is declared

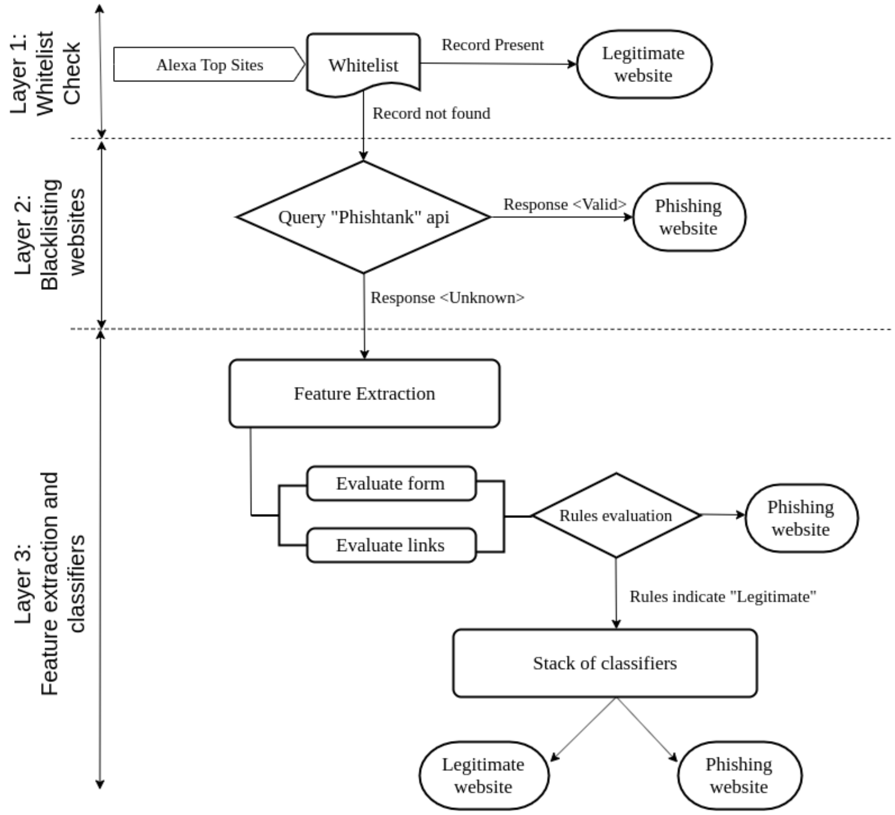

Fig. 6. System Architecture of the proposed solution

as legitimate as it doesn't attempt to capture any information. If a form is present, the request is passed to the next phase.

Phase 2: Hyperlinks extraction and analysis

In this phase, the hyperlinks are extracted from the requested webpage and analyzed according to the following rules:

- Any genuine website will have at least one hyperlink. The static informative websites also have a few hyperlinks. Thus, if any website doesn't have any hyperlink, it is declared as phishing.

- Attackers exploit the vulnerability of using "null pointer" in hyperlinks to give an illusion of having hyperlinks on the webpage. Any genuine website will never have a null pointer in hyperlinks and thus any webpage having them is declared as phishing.

- Evaluating the hyperlinks extracted from the webpage: In a genuine website, only a few hyperlinks point to any other domain and rest all hyperlinks point to the original domain of the webpage. The ratio of links pointing to the same domain vs other domain is evaluated and if the outer domain links are beyond a threshold, it is declared as phishing. From literature [30], it has been found that a 36% threshold of webpages pointing to a foreign domain is most effective in finding the phishing websites and hence this value is used.

The result of this phase if declares the webpage as phishing, the execution stop, and the user is given a prompt alert. If the result shows the website as legitimate, we don't rely on and further process the request to the next layer to be very sure of the result.

Phase 3: Stack of classifiers

In the experiments performed by various literature, it has been found that machine learning classifiers tend to give accurate results but consume too much time and computational cost. This is the reason, the request is passed to this layer only when all other possibilities to decide the status of the webpage have been tried that could give a fast response. In other words, the requests coming to this layer are new websites that have not been encountered yet. Three best classifiers based upon the high accuracy performance provided by each of them when applied to the phishing dataset have been chosen. Random Forest, SVM, and Logistic Regression have been chosen in this approach. If a single classifier is used, chances of true positives and true negatives are high whereas the negative detection results are exempted when a combination of three classifiers is used.

In this approach, every classifier is trained with a phishing website dataset available from UCI machine learning repository [33] which contains 11055 records having 30 features. For optimum results, only the best parameters are chosen which are evaluated by Random Forest Classifier by removing 'Gini impurity' from the features. This involves pruning trees to that extent for a node where a subset of most important features is created. Evaluation is further discussed in the next section.

The prediction from all three classifiers by using the best features is combined to give the final prediction as if a

TABLE I. RULES FOR CLASSIFICATION APPLIED ON THE FEATURES OF WEBSITES

Rules	Feature	Condition
	1. Address Bar based features	
1	URL Having IP	Usually not present in legitimate websites
2	URL Length	Usually <54 observed for legitimate websites
3	Tinu URL	Check URL shortening services
4	Having @ symbol	Not allowed in legitimate websites
5	Double slash	Allowed only once (after http/https ://)
6	Prefix/Suffix	Domain Name shouldnt include (-) symbol for legitimate sites
7	Sub Domain/ Multi Sub Domain	Phishing if number of dots greater than 2 in domain part
8	SSL final state	Checks for https and compares certificate issuer with trusted issuer list. Age of certificate should be greater than 1 year
9	Domain Registration	Checks updated date and expiration date of domain to find out vailidity. If expires <1year – Phishing
10	Favicon	Favicon loaded from foreign domain means Phishing site
11	Non-standard ports	Check status of common ports
12	HTTPS token	Checks if attacker is tricking the user by putting https in domain part
	2. Abnormal based features	
1	Request URL	Checks the requested URLs outside the webpage. Phishing if >61%
2	URL of Anchor	Average of hyperlinks pointing to foreign domain (<31% for legitimate website)
3	Links in tags	Average numer of meta, link and script tags should be <17% for legitimate
4	Server Form Handler	If SFH is “about: blank” or empty – phishing
5	Email submit	Checks usage of ‘mailto:’
6	Abnormal URL	If hostname is not included in URL – Phishing
	3. HTML and JavaScript based features	
1	Website forwarding	website redirection >4 – Phishing
2	Status bar customisation	if “onMouseOver” event changes status bar – Phishing
3	Right Click	If disabled – Phishing
4	Pop-up window	If pop-up window asks for user credentials input – Phishing
5	IFrame Redirection	If using IFrame without “frame orders” – Phishing
	4. Domain based features	
1	Age of Domain	Finds out how old the URL is. If >6 months – Legitimate
2	DNS Record	If claimed identity not recognised by WHOIS or no records found for hostname or DNS record is empty – Phishing
3	Website Traffic	Website Rank >10000 in Alexa database – Phishing
4	PageRank	If <0.2 – Phishing
5	Google Index	Webpage indexed by Google – Legitimate
6	Links pointing to page	If 0 – Phishing

particular classifier fails to identify the phishing website, the rest of the classifiers can help.

V. RESULTS AND EVALUATION

The proposed solution in the previous section is available in the form of a browser extension that has been tested in Google Chrome and Firefox web browser. The extension is developed in Python and is browser-independent. To test the performance and check the accuracy of this browser extension, a dataset containing random 1000 legitimate URLs and 1000 phishing URLs is being used and the achieved accuracy for classifying these web URLs accurately is found to be 98.1%.

The proposed browser extension has two execution layers:

- After the requested URL is not found in the whitelist store, then execution passes to the layer where PhishTank API is queried for the URL verification. This process saves time in processing the URLs that are already known to be phishing and hence the fast response to the user.

- The next layer is based on feature extraction where the website data is pre-processed and features are extracted. It checks for the presence of any form in the website and evaluates hyperlinks in the website based upon the rules defined in the previous section. If a website is declared phishing here at this phase, the execution stops. If it is declared suspicious, we further process the next phase.

- The final phase consists of a stack of three pre-trained classifiers. For choosing the best important features to achieve optimum performance, the best parameters need to be selected out of 30 features related to website data. For this purpose, Random Forest Classifier is used by choosing ‘Gini’ criteria. Fig. 7 shows the graph of the importance of each feature.

For testing the effectiveness of the proposed solution, the same dataset has been evaluated on individual classifiers, namely, Logistic Regression, SVM, Random Forest and a stack which combines prediction of these three. Observations of the experiments are described below and summarized in Table II:

- Fitting logistic regression and creating confusion matrix of predicted values and real values on the Phishing website dataset, 92.3% accuracy is observed.

- Using support vector machine with an rbf kernel and using “gridsearchcv” to predict best parameters for SVM turns out to be a really good choice, and fitting the model with predicted best parameters, 96.47% accuracy is attained.

- Using “gridsearchcv” in Random Forest for feature importance to get the best parameters and fitting best parameters, accuracy 97.26% is achieved.

- Using combined prediction of stack of 3 classifiers (Logistic Regression, SVM and Random Forest classifiers) turns out to be the most effective solution achieving accuracy of 98.1%.

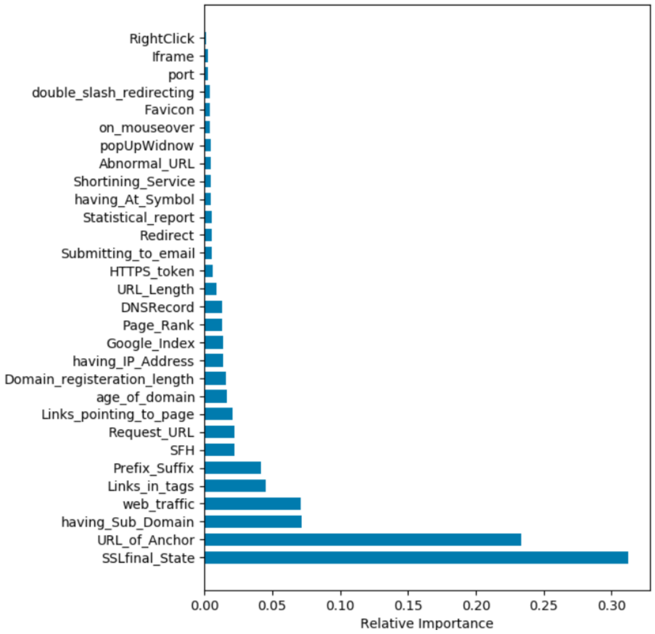

Fig. 7. Relative Importance of features evaluated through Random Forest classifier using GridSearch for Websites features

TABLE II. PERFORMANCE ANALYSIS OF CLASSIFIERS COMPARED TO PROPOSED FRAMEWORK FOR CLASSIFYING WEBSITES AS LEGITIMATE AND PHISHING CORRECTLY

	Accuracy	Mean Squared error	Average execution time(ms)
Logistic Regression	92.3%	0.864	0.98
SVM	96.47%	0.041	0.87
Random Forest	97.26%	0.051	1.75
Proposed framework	98.1%	0.047	0.62

VI. CONCLUSION

Phishing attack is a cyber-security issue that has been prevailing the internet for almost 3 decades now and still, people are getting phished daily. Attackers befool internet users with new phishing techniques and steal their personal secure information. Mitigation of phishing attacks consists of Detection, Prevention and Correction approaches. The phishing problem is very wide and no single solution exists to mitigate the entire vulnerabilities properly. Phishers find out vulnerabilities in the existing solutions and come up with new attacks. Prevention from getting Phished is better safety measure and user's awareness may help in protecting themselves from becoming phishing victims. User awareness helps them in analyzing the website or email in first look based on its prominent features. Software-based anti-phishing solutions to defend the user from email and website phishing are discussed in detail in this paper. Multiple techniques are often combined and implemented to mitigate specific attacks. Automatically detecting phishing web pages has attracted much attention from researchers. Several approaches and comprehensive strategies have been suggested to tackle phishing. This research paper provides a basic understanding of phishing attacks, its life cycle, and popular attack techniques. The advantages and disadvantages of every solution are discussed which helps in analyzing and choosing the appropriate mechanism for implementation. This helps in finding the vulnerabilities in each solution and a direction for future research through modification in these solutions.

After a thorough analysis of existing anti-phishing methodologies, a new browser extension is proposed in this paper which combines the advantages of profile matching techniques and machine learning classifiers. The main motive is to provide the result to the user as fast as possible. Thus, the solution should be efficient to give accurate results in less time. The proposed framework utilizes the resources available from reliable sources to speed up the process. The trusted agencies like 'Alexa' are used for creating whitelist having legitimate websites and 'PhishTank' is used for blocking phishing websites already researched by them. This way, most of the website requests get responses in a very short time. In the case of new websites, that may cause zero-day attacks, the stack of classifiers utilizes the best features and evaluates the website's extracted features to classify it into phishing or legitimate. This approach is capable to classify websites correctly with 98.1% accuracy.

REFERENCES

[1] K. Krombholz, H. Hobel, M. Huber, and E. Weippl, "Advanced social engineering attacks," *Journal of Information Security and applications*, vol. 22, pp. 113–122, 2015.

[2] "Phishing activity trends report 1 quarter 2019." https://docs.apwg.org/reports/apwg_trends_report_q1_2019.pdf. Accessed: 11-10-2019.

[3] "State of the phish 2019 report." https://info.wombatsecurity.com/hubfs/Wombat_Proofpoint_2019%20State%20of%20the%20Phish%20Report_Final.pdf. Accessed: 14-10-2019.

[4] "Global phishing survey: Domain name use and trends in 2016." https://www.antiphishing.org/resources/apwg-reports/domain-use-and-trends. Accessed: 06-07-2019.

[5] "Internet security threat report, symantec volume 22." https://www.symantec.com/content/dam/symantec/docs/reports/istr-22-2017-en.pdf. Accessed: 11-09-2019.

[6] "Must-know phishing statistics 2017." https://blog.barkly.com/phishing-statistics-2017. Accessed: 14-08-2019.

[7] P. Khurana, A. Sharma, and P. K. Singh, "A systematic analysis on mobile application software vulnerabilities: Issues and challenges," *Indian Journal of Science & Technology*, vol. 9, no. 32, pp. 1–6, 2016.

[8] "Google safe browsing." https://developers.google.com/safe-browsing/. Accessed: 06-07-2019.

[9] "Phishtank." https://www.phishtank.com/. Accessed: 06-07-2019.

[10] T. M. Mitchell *et al.*, "Machine learning. 1997," *Burr Ridge, IL: McGraw Hill*, vol. 45, no. 37, pp. 870–877, 1997.

[11] R. S. Ludwig and M. J. Piovoso, "A comparison of machine-learning classifiers for selecting money managers," *Intelligent Systems in Accounting, Finance & Management: International Journal*, vol. 13, no. 3, pp. 151–164, 2005.

[12] J. R. Quinlan, "Induction of decision trees," *Machine learning*, vol. 1, no. 1, pp. 81–106, 1986.

[13] J. R. Quinlan, *C4. 5: programs for machine learning*. Elsevier, 2014.

[14] A. C. Lorena, L. F. Jacintho, M. F. Siqueira, R. De Giovanni, L. G. Lohmann, A. C. De Carvalho, and M. Yamamoto, "Comparing machine learning classifiers in potential distribution modelling," *Expert Systems with Applications*, vol. 38, no. 5, pp. 5268–5275, 2011.

[15] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[16] V. Vapnik, *The nature of statistical learning theory*. Springer science & business media, 2013.

[17] E. Kirda and C. Kruegel, “Protecting users against phishing attacks with antiphish,” in *29th Annual International Computer Software and Applications Conference (COMPSAC’05)*, vol. 1, pp. 517–524, IEEE, 2005.

[18] P. Prakash, M. Kumar, R. R. Kompella, and M. Gupta, “Phishnet: predictive blacklisting to detect phishing attacks,” in *2010 Proceedings IEEE INFOCOM*, pp. 1–5, IEEE, 2010.

[19] A. Belabed, E. Aïmeur, and A. Chikh, “A personalized whitelist approach for phishing webpage detection,” in *2012 Seventh International Conference on Availability, Reliability and Security*, pp. 249–254, IEEE, 2012.

[20] Y. Zhang, R. Jin, and Z.-H. Zhou, “Understanding bag-of-words model: a statistical framework,” *International Journal of Machine Learning and Cybernetics*, vol. 1, no. 1-4, pp. 43–52, 2010.

[21] “Cosine distance.” http://reference.wolfram.com/language/ref/CosineDistance.html. Accessed: 22-11-2017.

[22] R. Islam and J. Abawajy, “A multi-tier phishing detection and filtering approach,” *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 324–335, 2013.

[23] A. A. Akinyelu and A. O. Adewumi, “Classification of phishing email using random forest machine learning technique,” *Journal of Applied Mathematics*, vol. 2014, 2014.

[24] R. Gowtham and I. Krishnamurthi, “A comprehensive and efficacious architecture for detecting phishing webpages,” *Computers & Security*, vol. 40, pp. 23–37, 2014.

[25] R. S. Rao and S. T. Ali, “Phishshield: a desktop application to detect phishing webpages through heuristic approach,” *Procedia Computer Science*, vol. 54, pp. 147–156, 2015.

[26] A. K. Jain and B. B. Gupta, “A novel approach to protect against phishing attacks at client side using auto-updated white-list,” *EURASIP Journal on Information Security*, vol. 2016, no. 1, p. 9, 2016.

[27] “jsoup: Java html parser.” https://jsoup.org/. Accessed: 25-12-2018.

[28] Y. Xiang, Y. Li, Y. Bo, and Y.-f. LI, “Phishing website detection using c4. 5 decision tree,” *DEStech Transactions on Computer Science and Engineering*, no. itme, 2017.

[29] Y. Li, Z. Yang, X. Chen, H. Yuan, and W. Liu, “A stacking model using url and html features for phishing webpage detection,” *Future Generation Computer Systems*, vol. 94, pp. 27–39, 2019.

[30] Y. Ding, N. Luktarhan, K. Li, and W. Slamu, “A keyword-based combination approach for detecting phishing webpages,” *computers & security*, vol. 84, pp. 256–275, 2019.

[31] S. Maurya, S. Sharma, A. Singh, and A. Jain, “Investigation of cyber security practices in academic institutions,” in *5th International Conference on Cybercrime and Computer Forensics (ICCCF)*, IEEE, 2017.

[32] “Alexa.” https://www.alexa.com/. Accessed: 05-10-2019.

[33] “Phishing websites data set..” https://archive.ics.uci.edu/ml/datasets/phishing+websites. Accessed: 14-02-2019.