Journal of King Saud University – Computer and Information Sciences xxx (xxxx) xxx

Contents lists available at ScienceDirect

Journal of King Saud University – Computer and Information Sciences

journal homepage: www.sciencedirect.com

A predictive model for phishing detection

A.A. Orunsolu [a], A.S. Sodiya [b], A.T. Akinwale [b]

[a] Department of Computer Science, Moshood Abiola Polytechnic, Ojere, Abeokuta, Nigeria
[b] Department of Computer Science, Federal University of Agriculture, Alabata, Abeokuta, Nigeria

ARTICLE INFO

Article history:
Received 8 April 2019
Revised 12 November 2019
Accepted 13 December 2019
Available online xxxx

Keywords:
Anti-phishing
Cyber-attacks
Identity theft
Middleware
Spoofed pages
Threat

ABSTRACT

Nowadays, many anti-phishing systems are being developed to identify phishing contents in online communication systems. Despite the availability of myriads anti-phishing systems, phishing continues unabated due to inadequate detection of a zero-day attack, superfluous computational overhead and high false rates. Although Machine Learning approaches have achieved promising accuracy rate, the choice and the performance of the feature vector limit their effective detection. In this work, an enhanced machine learning-based predictive model is proposed to improve the efficiency of anti-phishing schemes. The predictive model consists of Feature Selection Module which is used for the construction of an effective feature vector. These features are extracted from the URL, webpage properties and webpage behaviour using the incremental component-based system to present the resultant feature vector to the predictive model. The proposed system uses Support Vector Machine and Naïve Bayes which have been trained on a *15*-dimensional feature set. The experiments were based on datasets consisting of 2541 phishing instances and 2500 benign instances. Using 10-fold cross-validation, the experimental results indicate a remarkable performance with 0.04% False Positive and 99.96% accuracy for both SVM and NB predictive models.

1. Introduction

Phishing is an online fraudulent act that uses social engineering and technical subterfuge to deceive Internet users and acquire their sensitive data or critical online information (Gowtham and Krishnamurthi, 2014; Gupta et al., 2016). Social engineering techniques intend to acquire unsuspecting users' identity or sensitive confidential information through the use of spoofed emails, fake websites, dubious online adverts/promos, fake SMS from service providers or online companies, spear phishing etc. The common targets in social engineering schemes include big corporations, financial institutions, payment companies, military and government agencies who usually suffered huge financial and brand credibility damages (APWG security report, 2017). For instance, Stats and Trends, 2017 security reports indicated that nearly about $5 billion were lost between October 2013 and December 2016 affecting more than 24,000 victims worldwide in a W-2 type of phishing attack. The W-2 phishing emails have been reported to be the most dangerous phishing email scams in recent times as its goal is to file fraudulent tax returns and claim refunds.

On the other hand, technical subterfuge schemes usually involve the use of malicious software or crimeware which is usually installed on a computer or its associated devices without the knowledge of the victim (Khonji et al., 2013; Gupta et al., 2016). Some techniques used in technical subterfuge include DNS poisoning, keyloggers, session hijacking, host file poisoning, content injection etc. In recent times, phishers have developed "*ransomware*" which executes a malicious code that adversely affects computing resources and demands a ransom payment to restore the resources to the original state. The incidence of these ransomware-based phishing emails as reported by CSO, showed that 93% of phishing emails are now "*ransomware*". The report observed that most victims tend to pay quickly because of the sensitive nature of their resources (CSO Online report, 2016).

In response to the phishing menace, various countermeasures called Anti-Phishing System (APS) were developed. However, phishers continue adopting evolving new sophisticated patterns to defeat current defence systems. Specifically, most existing APS have problems of the possibility of a zero-day attack, superfluous computational overhead, high false positive and negative rates (Chin, 2018; Moghimi and Varjani, 2016). While some existing

Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

E-mail address: orunsolu.abdul@mapoly.edu.ng (A.A. Orunsolu)

https://doi.org/10.1016/j.jksuci.2019.12.005
1319-1578/

Please cite this article as: A. A. Orunsolu, A. S. Sodiya and A. T. Akinwale, A predictive model for phishing detection, Journal of King Saud University – Computer and Information Sciences, https://doi.org/10.1016/j.jksuci.2019.12.005

methods innovated one or more features to achieve promising results (Moghimi and Varjani, 2016), others extracted a subset of existing feature corpus to achieved the same results (Adebowale et al. 2018). Nonetheless, several APSs using Machine Learning (ML) and data mining techniques achieved a promising accuracy rate peaking at 99.62% (Hota et al. 2018; Chin, 2018; Sonowal and Kuppusamy, 2017; Tan et al. 2017), the choice and the performance of the feature vector on these algorithms limit the effective detection system (Qabajeh et al., 2018). Motivated by this premise, in this work, we pursue a robust based anti-phishing scheme to overcome the current challenges facing existing APS by asking the following questions.

a. Is it possible to achieve a more significant detection accuracy result by selecting or combining existing feature corpus?
b. Can the extracted subset of the feature vector achieve comparable results on more than one ML technique?
c. Must new features be continually proposed to fight phish despite the large available phishing feature corpus ranging from visual to text-based?

These questions indicate the need to examine the possibility of using existing phishing feature corpus to build a robust anti-phishing scheme with significant efficiency. Our focus is to propose an efficient phishing "fingerprints" i.e. feature vector with significant detection accuracy across more than one ML algorithms. Specifically, we choose Support Vector Machine and Naïve Bayes for the evaluation of our feature vector because most existing APSs have used these ML algorithms more than others to benchmark their approach in most extant literature available (Adebowale et al., 2018; Tan et al., 2017; Gowtham and Krishnamurthi, 2014; Hota et al., 2018; Mao et al., 2019; Han et al., 2012). Although other ML such as KNN have been used in phishing problem, both SVM and NB have been found most suitable due to their binary classification attribute and simplicity (Anwar et al., 2017; Isa et al., 2008; Dhanalakshmi and Chellappan, 2013).

The proposed scheme makes the following contributions to anti-phishing research:

a. This paper proposes an efficient selection of a subset of large existing phishing "fingerprints" called feature vector. The selection technique is built using a feature frequency assessment of various feature sets. These features are selected from the URL, the web page's property and the webpage's behaviour due to their high-ranking preferences and promising discriminative attributes (Gupta et al., 2017; Aburrous et al., 2010). To the best of our knowledge, this is the first paper to examine such criteria in constructing a feature vector for the anti-phishing system. The main motivation for this approach is to determine how the existing phishing feature vector can be sufficiently integrated into an effective countermeasure.
b. The approach proposes an incremental component-based system to present the resultant feature vector from each filter (i.e. URL, Web properties or web behaviour) within the system to a predictive model using two different ML algorithms namely SVM and NB. The purpose of this is to provide practical solutions for managing scale and complexity within the proposed system. The main motivation for this is to prioritize managing of the feature set.
c. The approach uses experiments to demonstrate that the proposed approach achieved significant detection accuracy with agreeable runtime. This implies that an efficient APS is mainly dependent on discriminative features which can be intelligently extracted from existing feature vector without necessary innovating new features, though, such may be desirable. Besides, since phishing is a problem which has existed over a long time, an efficient APS needs to continuously keep up with historical "fingerprints" in proposing new solutions to maintain a smart relationship with the evolution of the phishing problem.

The remaining sections in this research article are presented as follows: Section 2 presents the literature review. The basics of the system architecture of our proposed methodology are discussed in Section 3. In Section 4, the implementation and evaluation of the proposed method are presented. Conclusions and future works are put forth in Section 5.

2. Literature review

The word "phishing" was first recorded in a Usenet newsgroup called AOHell on January 2, 1996, to describe the theft of users' credentials on America Online (AOL) by a group of hackers and since then, the scale and sophistication of phishing attacks have been on increase with huge financial and reputational damages on online users. As phishing attacks continue to ravage online community, various research efforts have examined reasons why people become victims of phishing attacks. In one of the earliest works, Dhamija et al. (2006) identified lack of computer system knowledge, lack of knowledge of security and security indicators; visual deception and bounded attention as reasons why people fell for phishing. Similarly, Mohammed et al. (2015) investigated why phishing still works in spite of all efforts at mitigating the menace. Their reports showed that people were still vulnerable at 53%, even when primed to identify phishing attacks. On the socio-demographic perception of security tips messages, Orunsolu et al. (2018) identified gender, academic qualification and user's computer knowledge as factors that can influence users' recognition of phishing messages.

In light of these weaknesses, several software-based approaches have been introduced to combat phishing attacks. These solutions range from simple list-based methods to the machine learning approach. For example, Han et al. (2012) developed an Automated Individual White-List (AIWL) in which the record of well-known benign sites visited by users is kept. The AIWL maintains a user interface information where the user inputs his or her details to prevent unhealthy disclosure of confidential information to malicious sites. This method provides an effective defence mechanism against pharming and dynamic phishing attacks. However, the method is dependent on how users train their browsers i.e. user-feedbacks dependent. This weakness still makes users, both experienced and amateur, susceptible to phishing if the level of training their browser is low. In related research, Jain and Gupta (2016) designed an auto-updated white list system for protecting users from phishing attacks. The approach which consists of domain IP addressing matching phase and links feature extraction phase provided fast access time and high detection rate of 86.02%. However, the false-negative rate of 1.48% is a limiting factor in a critical online transaction system.

Mao et al. (2019) presented an anti-phishing scheme based on aggregation analysis of webpage layouts using property vector extraction, property vector generation, comparison vector generation and machine learning classifiers. The approach was evaluated using phishing standard datasets from PhishTank with an accuracy above 93%. Similarly, Chiew et al. (2015) examined a method of detecting phishing pages using a logo image which consists of logo extraction and identity verification modules. In the logo extraction process, the logo image on a site is detected and extracted from all the downloaded image resources of a web page. This phase is followed by a Google image search process in which the portrayed

Please cite this article as: A. A. Orunsolu, A. S. Sodiya and A. T. Akinwale, A predictive model for phishing detection, Journal of King Saud University – Computer and Information Sciences, https://doi.org/10.1016/j.jksuci.2019.12.005

identity of the logo image is retrieved. In the end, the authors were able to establish the relationship between the query return by Google and the domain name to determine the status of the website.

Using webpage link information, Gowtham et al. (2017) presented a defence system in which all possible target domains are identified on a suspicious webpage without much reliance on the search engines. The method worked by visiting the links of a suspicious page up to level two to check for the possible number of domains that can be reached. A domain count value was defined to identify the technique in generating the Target Domain set and thereafter, a cost matrix is formulated based on the relationship that exists between the domains in the Target Domain set and the loading suspicious webpage. A Target Validation algorithm was then considered to define the correctness of the prediction. The system was evaluated with a dataset consisting of 3675 active phishing and legitimate websites with a true positive rate of 99.53% and a false-positive rate of 0.45%. However, the prediction of this approach is largely dependent on Target Domain construction, which can be infeasible on a large-scale deployment. Also, this approach is inefficient when used for phishing target detection when shortening service is deployed.

On the use of Stacking strategy to improve the performance of data mining techniques, Li et al., 2019 presented an approach in which 20 features consisting of URL features and HTML features were extracted. These features were trained using a stacking model consisting of Gradient Boosting Decision Tree, XGBoost and LightGBM. The approach was experimented using a large dataset with 98.60% accuracy and 1.54% false alarm rate. However, the average runtime of the stacking model was missing and as a result, the efficiency of the approach cannot be determined in a real-life implementation of the system.

On the use of phishing kits, Orunsolu et al. (2017) presented an anti-phishing kit scheme for secure transactions. Their method used an architectural approach that consists of a sorter module and signature detection module to present an effective defence system for phishing kit deployment in the proliferation of phishing attacks. The approach was evaluated using several web-cloning tools and generalized datasets from Alexa and PhishTank. The experimental results showed that the approach can mitigate phishing attacks effectively.

Based on heuristics approach, Hota et al. (2018) constructed an ensemble machine learning-based model to detect phishing attacks in an email by using Remove-Replace Feature selection techniques which reduces features from original feature space by randomly selecting a feature and remove such features if the accuracy associated with the feature is unchanged. The empirical results indicated an accuracy of 99.27% with only 11 features. In a similar approach, Sonowal and Kuppusamy (2017) presented an approach called PhiDMA, which incorporated five layers consisting of auto-whitelist layer, URL features layer, Lexical signature layer, string matching layer and accessibility score comparison layer. Their model is especially suited for persons with visual impairments and its empirical results indicated an accuracy of 92.72%. In the same vein, Zouina and Outtaj (2017) investigated a lightweight URL phishing detection system using SVM and similarity index on six features extracted from the domain address of a webpage. The system achieved an accuracy rate of 95.80%.

On the use of Fuzzy logic (FL), Aburrous et al. (2008) presented one of the earliest works on the use of this technique. The authors utilized fuzzy logic in a simulated phishing experiment by sending fake emails to employees of a particular bank in Jordan. Their purpose was to discover features that influenced users' judgement of phishing messages. In the end, the authors developed a Fuzzy logic classification model based on six criteria which were assigned values ranging from Phishy, Genuine to Doubtful. A new FL approach by Barraclough and Sexton (2015) used six inputs consisting of user behaviour profile, legitimate sites rules etc. to describe a neuro-fuzzy methodology. A total of 300 features were extracted from the six input categories to train and evaluate the FL inference system. However, the use of a large feature sets consisting of only text-based is an obvious limitation of this approach. To improve this work, Adebowale et al. 2018 developed an Adaptive Neuro-Fuzzy Inference System using integrated 35-dimensional features of text, images and frames for web-phishing detection and protection. These features were selected using Chi-Square Statistics and Information Gain technique was used to decrease the size of the feature set. The approach was experimented using 13,000 available datasets and evaluated using SVM, K-NN and ANFIS. This system achieved 98.3% accuracy. However, the average runtime time for the approach was still non-negligible and the expectation for a set of globally accepted visual features (i.e. their use of image features) on the majority of websites may still need to be addressed in the real-life implementation of the approach (Varshney et al., 2016a).

On the use of search engine-based approach, Dunlop et al. (2010) developed a technique called Goldphish. The method extracted the logo of the website and converted the logo using OCR technology into text which is then used as a query into Google. The approach achieved a True Positive Rate of 98%. However, the delay in rendering the webpage image into text is a limiting factor for real-time implementation of this approach. Similarly, (Varshney et al., 2016b) presented an optimized search engine-based technique called Lightweight Phishing Detector which used the combination of domain name and title to detect phishing attacks. The approach was experimented using 500 URLs from PhishTank and Alexa dataset. The work achieved TPR of 99.5%. However, the system cannot efficiently filter webpages based on language differences. To improve this approach, Jain and Gupta (2017) proposed a technique in which a web search query approach is combined with two-level authentication. The approach consists of the domain name and title extractor, search query formation, google web search lookup, two-level authentication and webpage prediction. Experimental results indicated that the method achieved 98.05% overall accuracy.

On SMS-based anti-phishing scheme, Shabtai et al. (2012) investigated a lightweight scheme for detecting host-based malware on a mobile phone. Their approach considered various features and events in the operation of a mobile device and then adopted machine learning to classify data as phishing or otherwise. Experimental results indicated that the approach is effective in detecting phishing attacks on mobile devices. In similar research, Bottazzi et al. (2015) developed a detection framework called MP-Shield for detecting malicious activities in mobile software and privacy data leakage in android applications. The approach was implemented as a proxy service using the TCP/IP stack. The tool provided an effective evaluation of android applications at large scale with high usability. Similarly, Sonowal and Kuppusamy (2019) developed a Phoneme based Phishing Verification Model for persons with visual impairments called MMSPhiD. The model consists of the machine learning-based approach, typosquatting based approach and the phoneme-based approach with the main focus of detecting phishing URLs and other related attacks. The model achieved a detection accuracy of 99.03% and provided a practical anti-phishing solution for persons with visual impairments.

The proposed predictive model in this work has adopted the ranking of feature selection for spam and phishing detection system (Toolan and Carthy, 2010; Gupta et al., 2017; Aburrous et al., 2010) in selecting the feature category used in our approach. In the end, we identified that URL property, webpage's behaviour and webpage's property as a relevant feature category to achieve our objective of an efficient phishing detection system. This is

Please cite this article as: A. A. Orunsolu, A. S. Sodiya and A. T. Akinwale, A predictive model for phishing detection, Journal of King Saud University – Computer and Information Sciences, https://doi.org/10.1016/j.jksuci.2019.12.005

because, individually these feature categories (i.e. URL, webpage's property or webpage's behaviour) has been used to achieve lightweight and efficient phishing detection system in most extant literature (Zouina and Outtaj, 2017; Li et al., 2019; Hota et al., 2018). Besides, these feature categories have found a greater influence in the design of anti-phishing schemes than other categories such as visual similarity, search engine-based approach etc. (Sonowal and Kuppusamy 2019; Qabajeh et al., 2018). More importantly, these feature categories have provided counter-attacks strategy for common phishing related attacks such as typosquatting, pharming attacks, ransomware etc. (Sonowal and Kuppusamy, 2019; Li et al., 2019; Moghimi and Varjani, 2016). The detection accuracy achieved by these feature categories, individually or collectively, is also promising. For instance, MMSPhiD (2019) which used a machine learning approach on URL and webpage attributes achieved accuracy of 99.03%. Therefore, the motivation for the integration of these existing feature category is to achieve a better anti-phishing detection system. Our approach is better than most existing APSs because we avoided features that may increase the runtime of the system. For instance, the image feature approach used in Adebowale et al., 2018 usually come with complicated image extraction process before web status can be computed (Jain and Gupta, 2017). Besides, the scale-dependent information of most existing favicons and logos is a limiting factor. Hence, the objective of this is to achieve Computational Efficiency (CE) in our approach. Also, since our approach is based on the integration of existing phishing feature corpus, we have introduced incremental construction of the component-based system to manage scale in the system development. Hence, this gives us the objective of Robusticity i.e. ability to resist total failure despite the unavailability of certain features due to design requirement or external influence. Besides, the incremental construction approach can be expanded to incorporate new features or delete redundant features without affecting the entire system structure due to the atomic and composite nature of the system's construction. Hence, this gives us the objective of Ease of Upgrade (EU). Table 1 presents a comparison of our approach with other existing literature concerning these design objectives.

3. The proposed system

The problem definition of phishing attack is a typical case of binary classification problem as an online communication (e.g. website or email or e-chat) is either Phish or benign. More formally, let w be a request that needs classification i.e.

$$w \xrightarrow{X} \{phish, benign\} \tag{1}$$

Then X is the anti-phishing system that takes features, $f_i \in w$ such that

$$w = \sum_{i}^{n} f_i n > 0 \; i.e. w \; is \; non - empty \; set \tag{2}$$

Table 1
Comparison of related works with our approach.

Work	CE	Robusticity	EU
Sonowal and Kuppusamy, 2017	Yes	No	No
Zouina and Outtaj 2017	Yes	No	No
Moghimi and Varjani, 2016	Yes	No	No
Li et al. 2019	Yes	No	No
Adebowale et al. 2018	Yes	No	No
Hota et al. 2018	Yes	No	Yes
Our Approach	Yes	Yes	Yes

Thus, a request contains at least one feature (e.g. links, HTML tags, scripts, SSL certificate etc.) on which prediction of its status can be queried or classified. Because these features can range from simple to complex, the proposed model uses feature frequency assessment for feature vector composition depicted by $X = \{x_1, x_2 \cdots x_n\}$ which assign label y to each $f_i \in w$, such that the label y is a binary class represented as:

$$y = \begin{cases} 1 \; (i.e. \; phishing) \\ 0 \; otherwise \; (i.e. \; genuine \; page) \end{cases} \tag{3}$$

Represented as

$$x_i : f(w) \rightarrow y \tag{4}$$

Equation 1 depicts the classification problem where, given a training data D, which contains $(w_1, w_2 \cdots, w_n)$ and each w_i contains a set of features $(f_1, f_2 \cdots, f_m)$. Also, the training data is a set of classes $C = (C_1, C_2)$ which represents phishing and legitimate sites such that:

$$C_I = \{w_i, f_i | w_i \in D, y = benign, i = 1 \cdots m\} \tag{5}$$

$$C_2 = \{w_i, f_i | w_i \in d, y = phishing, i = m + 1, \cdots p\} \tag{6}$$

Thus, each case $w_i \in D$ may be given a class $c_i \in C$ and is represented as a pair $(w_i, (c_i))$ where c_i is a class from C associated with the case $w_i$in the training data. Let H denote the set of classifiers for $D \rightarrow C$, where each case $c_i \in C$ is given a class and the goal is to find a classifier$h_i \in H$ that maximizes the probability that $h(c_i) = c$ for each test case. In the proposed system, two most common machine learning classifiers for phishing classification namely, Naïve Bayes and Support Vector Machine are chosen to investigate the performance of the feature set/vector and to maximize the accuracy of our proposed approach.

3.1. Feature selection module

A feature extraction process involves the identification of certain features which are characteristics of a particular set of data e.g. spam or phishing or benign etc. Such features are usually marked "fingerprints" of classes where they occur with less or no probability of occurring outside the known-set. In most cases, such features are usually mutually exclusive of the other classes. Numerous features have been extracted by past literature but the problem of the most representative features remain a concern. In this approach, a feature assessment based on frequency analysis of several features collected from research datasets and extant literature is used. This is defined as a Feature Selection Module (FSM) which consists of:

i The URL features
ii The Web document properties
iii The behaviour of the webpage

These three components are regarded as a filter in FSM and each filter is organized into the approach to make a system using incremental construction of a component-based system in Lau et al. (2012). Based on this approach, the three filters are built as a unit filter and composite filter to incrementally achieve an efficient detection approach. Where the unit filter consists of the heuristics defined in each component (computation unit) and its prediction score (invocation connector) which are jointly "fired" to the next unit filter, the composite filter represents the aggregate of all the unit filter from the system i.e. $S_0 \subseteq S_1 \subseteq S_2 \cdots \subseteq S_C$ where S_C is the composite filter built from the subsets of other units. Thus, it is the composite filter that computationally attached the system to the classification algorithm. Fig. 1 presents the architecture of

Please cite this article as: A. A. Orunsolu, A. S. Sodiya and A. T. Akinwale, A predictive model for phishing detection, Journal of King Saud University – Computer and Information Sciences, https://doi.org/10.1016/j.jksuci.2019.12.005

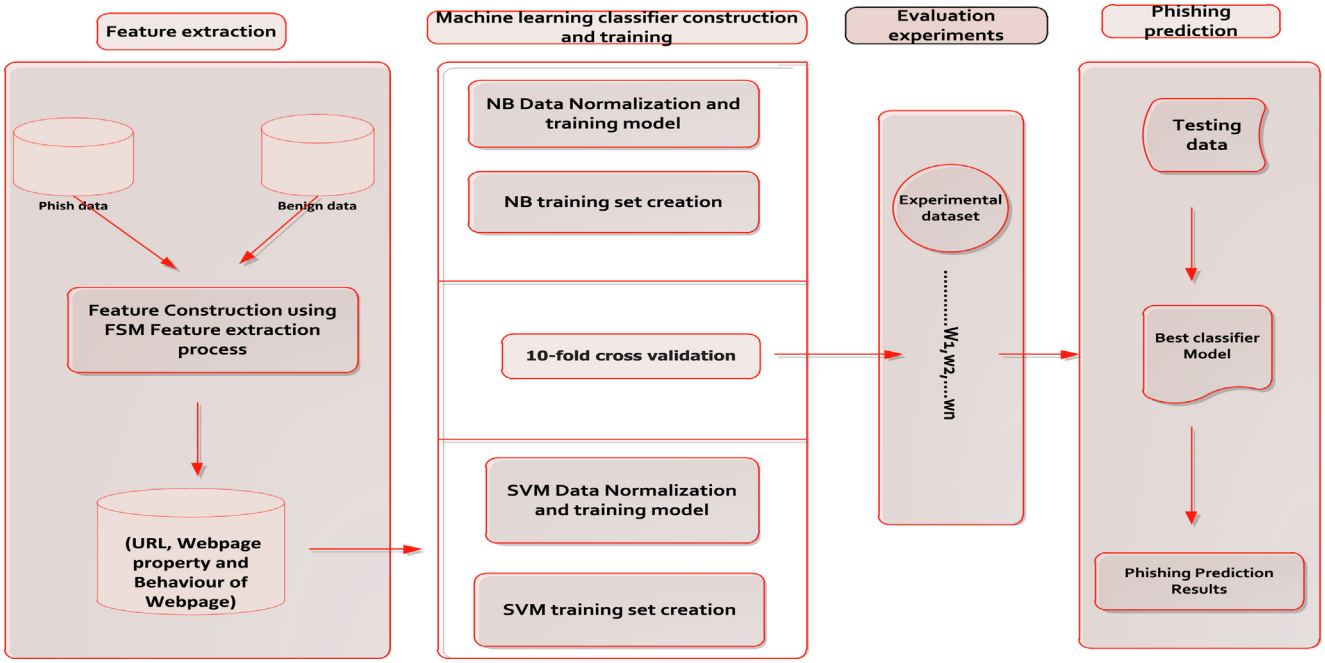

Fig. 1. A Robust Phishing detection system.

the proposed approach which consists of the feature extraction phase, machine learning classifier construction and training, evaluation phase and phishing prediction phase. The feature extraction phase takes mixed data as input from where features are extracted from the URL, Web document properties and web behaviour attributes.

For URL features, 5 features are extracted. In the web document features, another 5 features are extracted. The web behaviour attributes also consist of 5 features which are extracted to enhance multiple pages phishing detection. Although some other features are still available in these feature categories, we specially choose those features because the omitted one can be deduced from the chosen ones. For instance, the number of dots in most phishing URLs are associated with elongated domain names. Then, a target label is assigned to the extracted features. The machine learning classifier phase used the extracted features to train the chosen ML algorithms (i.e. SVM and NB). The evaluation phase is used to benchmark the performance of the proposed system through some experiments on standard datasets. In the end, the phishing prediction phase is used to determine the accuracy of the proposed model.

3.2. The URL features (F1-Filter)

The URL features represent the characteristics associated with web addresses where a particular page can be retrieved from the Internet. Phishers usually manipulate legitimate URLs in different ways to deceive unsuspecting users. The URL features are extracted either an absolute URL or relative URL by analyzing the links structure in the DOM. For URL identity extraction, the FSM considers the "*href*" and "*src*' attributes of the anchor links, particularly < a>, <area>, <link>, <img>, and < script > tags from the DOM tree of a webpage where web addresses are usually located. To extract the URL-based features, FSM uses downloaded PhishTank database consisting of confirmed phishing URLs that totalled 23,769. Based on the preliminary study of this database, FSM constructed a number of a query on certain URL characteristics selected from the existing researches (e.g. Zouina and Quttaj, 2017; Aburrous et al., 2010; Sonowal and Kuppusamy, 2017; Gowtham and Krishnamurthi, 2014) to determine their frequency on the list. Besides, we extended the query of FSM to the URL behaviour on the precompiled downloaded Alexa data corpus consisting of 1 million confirmed legitimate URLs to validate the correctness of the selected features from legitimate data sources.

Hence, the system is design based on several URL behaviours that differentiates a malicious page from the legitimate page (Moghimi and Varjani, 2016). For example, our query on the use of "@" symbol in a URL path returns no match for legitimate URLs even though the number of its occurrences skewed lower compared with other features used in our work. Despite this result, FSM chooses to include the symbol because of its unknown occurrence to benign URLs. Therefore, this feature provides a marked support vector plane of the marginal distance between phishing URLs and benign URLs.

Based on the methodology of frequency feature assessment from the two data sources, algorithm 1 is presented. Given an initial feature list $F_{URL(n)}$, the algorithm only selects features found in the two data sources. Then, the frequency of occurrence of the selected feature is then determined using equation (7). If this value exceeds the exclusion limit, the feature is included in the new feature list, *S*. This process is repeated until the initial feature list is exhausted. The new feature list, *S*, is then ranked and the performance of each feature is determined. Then, a final new feature of dimension, *m*, consisting of ranked best performing features are selected. The URL features selected in our approach used a Frequency Information (FI) whose values lie between 1 and 0. This value describes the statistical weight of each feature on the entire database. That is,

$$FI = F_{url} / \sum DB \tag{7}$$

$$0 \ll FI \ll 1 \tag{8}$$

where 0 means no occurrence found and 1 means found in all occurrences. Table 2 illustrates the frequency rate of the selected URL features and Table 3 presents the meaning of notations used in algorithm 1. After a close observation on the FI, FSM constructs

Please cite this article as: A. A. Orunsolu, A. S. Sodiya and A. T. Akinwale, A predictive model for phishing detection, Journal of King Saud University – Computer and Information Sciences, https://doi.org/10.1016/j.jksuci.2019.12.005

Table 2
The frequency of selected URL features set.

S/N	URL feature	Frequency rate
1	@ symbol in URL	0.23
2	IP based URL	0.50
3	URL with hexadecimal code	0.45
4	Long URL length	0.68
5	Multiple slash in the URL path	0.82

Table 3
List of notations and their meanings.

Notations	Descriptions
n	Number of features for frequency analysis
F_{url_i}	An instance of URL feature
d_{ph}	A database of confirmed phishing URLs
d_{be}	A database of confirmed legitimate URLs
θ	The threshold for feature analysis
p	Size of the dataset in Mendeley Desktop Application
s_i	Feature category for F2 & F3 feature
H_P	High impact phishing features
CFS (s)	Correlation Feature selection Function for s
f	An instance of feature in a particular feature category
t	Correlation factor e.g. symmetry uncertainty or correlation Pearson's coefficient
a	Counter
$f(s)$	The feature set of selected high impact features
$m(fs)$	The subset of high impact features selected for phishing detection

the value for each existing URL features and at the end, the following URL features were selected in our approach as their FI exceeded the exclusion limit defined within the system. A simple illustration of the exclusion limit (i.e. threshold value,θ) is given as:

$$1.00 << FI_{phishing} << 0.10,\ 0.00 << FI_{benign} << 0.20 \quad (9)$$

i. URL with @ symbol: This involves using @ symbol in the URL path of a website. This symbol is used to redirect traffic to phishing site whose domain name immediately followed the @ symbol. For example, HTTP// mapoly.edu.ng@gatewaypoly.edu.ng will direct a user to Gatewaypoly instead of Mapoly. The @ symbol usually comes with a shorter domain name unlike some other symbols such as "-" or ".".

Therefore, if URL contains @ symbol then phishing otherwise legitimate

ii. Using the IP address as URL: This involves the use of IP address to represent the domain name of a website. Usually, this practice is very common for hiding the original information of a domain name. Hence, such IP address usually denotes phishing or suspicious domain.

If URL contains Domain path as IP address then Phishing, otherwise legitimate

iii. URL with hexadecimal character code: Phishers usually hide phishing URLs by using hexadecimal codes to represent the numbers in the IP address. Each hexadecimal code usually begins with a "%" symbol. For instance, http://donefe.000webhostapp.com/auto/auto%20ferify/mail.php which was reported in January 2018 by PhishTank used the hexadecimal character code.

If URL contains hexadecimal character then phishing, otherwise legitimate

iv. URL Length: This involves getting a URL length that is more than 35 characters. For example, HTTP// womenincoachingsuccess.com found on the Alexa database is a legitimate URL. A close observation of the Alexa database indicated that any length of more than 35 is likely to be phishing.

If URL length greater than 35 then phishing Otherwise, legitimate

v. URL with multiple "//": This involves the use of more than one "//" in the domain name path of a URL. A search query on the 1 million Alexa database of legitimate URLs in a.csv excel format returns 0 for this feature.

If URL contains multiple "//" then phishing Otherwise legitimate

It should be noted that certain URL features such as the number of dots and use of "-" were omitted. This is because we observed that such features are usually related to the length of the URL. That is, the number of dots usually elongates the length of the URL. For instance, https://upgrade-identity.000webhostapp.com/recovery-checkpoint-login.html reported in January 2018 by PhishTank had elongated URL length with more than three number of dots. Besides, we did not include "-" because this character has about 11% (i.e. approximately 110,120) occurrences in the 1 million Alexa database. This size is even more than most testing and training data corpus found in the most anti-phishing literature. This is significant since the Alexa list contains the top most visited URLs on the web which are most of the times the target of a major phishing campaign. Moreover, the dash symbol is often associated with elongated URLs as observed from our query run on the phishing data corpus. Hence, our choice of the URL length to the best of our knowledge is sufficient to accommodate these omitted features.

Algorithm 1: URL Feature Assessment Frequency Analysis

Input: Updated Phishing Corpus, d_{ph}, Alexa top URL, d_{be}, Predefined threshold value,θ.

Output: URL-based Feature vector of dimensionS_m

Begin

For $i = 1$ *to n do begin*
$F_{URL(n)} \leftarrow$the set of all n URL features
IF $F_{url\_i} \in d_{ph}.OR.d_{be}$ **Then**
  $S \leftarrow$ new Feature List
Calculate the frequency of$F_{url_i} \in d_{ph}.d_{be}$
Calculate Frequency Information, FI, ofF_{url_i}
  {IF $FI_{F_{url\_i}} > \theta$, **Then**
    Append F_{url_i} to S
    Else Reject F_{url_i} **}**
    Next i
    Continue
      Rank$F_{url_i} \in S$
      Select top F_{url_i} features $\in S$
      Get performance measure of the$S$
    Identify the best performance measure as the best features
$S_m \leftarrow$ *bestfeatures*
End

3.3. The web document properties (F2-Filter)

The Web document properties of a webpage are extracted from Document tag which includes the Title tag, Meta tag, Alt attribute of tags, Title attribute of tags, meta description etc. where keywords associated with a web page's product or services are defined. Thus, a web documents property can be acquired from its keyword's identity. This extraction process is based on the concept of Term Frequency-Inverse Document Frequency (TF-IDF) method. The method is used to extract a set of keywords from document d (i.e. in TF-IDF, a webpage is treated as a document) which is collected from various portions of a webpage. The TF-IDF reflects the numerical statistic of how relevant a feature is to a document in a data corpus. This term is usually employed in information retrieval/data mining as a weighting factor. The TF-IDF value increases proportionally to the number of times a feature appears

Please cite this article as: A. A. Orunsolu, A. S. Sodiya and A. T. Akinwale, A predictive model for phishing detection, Journal of King Saud University – Computer and Information Sciences, https://doi.org/10.1016/j.jksuci.2019.12.005

in the document but is offset by the frequency of the feature in the corpus (Wikipedia, 2018). Thus, a particular term defined as t has a high TF-IDF weight if the term has a high term frequency in a given document D and a low document frequency if the term is relatively uncommon in the documents.

Given a document d and its term identity set t, the FSM uses the frequency rate measurement (i.e. frequency assessment analysis) to determine the inclusion of a feature into the discriminative feature class. The discriminative feature class is generated by marking the feature with the highest frequency rate from a number of features collected from previous works (e.g. Aburrous et al. 2010; Moghimi and Varjani, 2016; Toolan and Carthy, 2010; Zouina and Outtaj (2017); Hamid and Abawajy, 2014) using Mendeley Desktop Application Library as the feature repository. Algorithm 2 presents the flow of the system methodology while Table 3 depicts the list of notations and their meanings as used in algorithm 2. These works are selected because of their detection and evaluation rate returned promising results on True Positive (i.e. between 80 and 99%) with negligible False positive (<1%). Also, since the nomenclature representations of various features can be differently presented in various forms, there is a need to ensure efficient feature-to-feature inter-correlation. Based on this requirement, a Correlation-based Feature selection evaluation function is introduced. This correlation-based heuristic evaluation is represented as:

$$M = \frac{k.a}{\sqrt{k + k(k-1).b}} \tag{10}$$

Where M is the feature merit of feature subset containing k features, a is the mean feature-class correlation and b is the average feature-feature inter-correlation. The numerator of 1 indicates how predictive of the class a set of features are and the denominator provides how much redundancy there is among the features. This makes the approach to be computationally light and avoid overfitting in the feature selection method (Chandrashekar and Sahin (2014)). Thus, this is important in identifying how significant a particular feature is.

Based on this numerical evaluation, the FSM considers the following features ($F_{V2} = F_1, F_2 \cdots F_5$.) where value 0 indicates non-phishing status and >0 indicates suspicious or phishing status. Although several features satisfy the greater than zero condition, we choose features that have more than 20% occurrence (Threshold value). Table 4 presents the frequency analysis of the selected features for both web documents properties and web behaviour attributes.

i. **Domain name check**: In most usual cases, website domain names ($\boldsymbol{D_n}$) have a strong relationship with their contents ($\boldsymbol{C}$) depicting the nature of products or services offered by the webpage. The keywords in this domain name are usually part of the base domain URL and should form the label for most links/anchors on the page. Therefore, if the keyword identity set of a page is not related to its contents (at least 70%), then it is phishing. Otherwise, it is legitimate.

$$F2_1 = \begin{cases} 0\, C \subseteq 0.7 * D_n \\ >0\, C \not\subset 0.7 * D_n \end{cases} \tag{11}$$

ii. **The domain name in the path of a URL:** Some phishing URLs add the domain name of a legitimate website within the path segment of a URL in an attempt to scam users into believing that they are dealing with an authentic website. This implies that this feature can equally detect the use of prefix or suffix by phishers in reshaping suspicious domain name as the genuineness will be low due to inappropriate keyword identity set. Therefore, if the domain name in the path of a URL contains prefix or suffix ($D_{\mathbf{ps}}$) that is not indicated in its contents then it is phishing. Otherwise, it is legitimate.

$$\boldsymbol{F2}_2 = \begin{cases} 0\, \boldsymbol{D_{ps}} = \boldsymbol{D_{url}} \\ >0\, \boldsymbol{D_{ps}} \neq \boldsymbol{D_{url}} \end{cases} \tag{12}$$

iii. **Server Form Check (SFC)/Pop-Up Window:** In a normal form processing operation, the domain name of a webpage is the same as the active form field address where the information is processed. But if there are any discrepancies between these two addresses or the domain name of the form is empty or missing, then it is likely to be phishing. Besides, a pop-up window can be activated by the phisher to circumvent this attribute. Since the goal of every phishing web page is to have access to user's details, and they achieve this by sending the user's form field to their servers where they can have access to it through a pop-up window. Although most modern browsers allow *window. open* (i.e. one of the commands for creating pop-up window) to run only if it was called by user interaction, phishers can trigger on a mouse click event listener attached directly to the web documents to achieve their malicious intent. In this way, the call restriction to the mouse click events can be hijacked. Therefore, if a webpage contains a Pop-up window and the domain name/keyword identity set on the pop-up window is not related to the foreground URL, then it is phishing. Otherwise, it is genuine.

$$F2_3 = \begin{cases} 0\, Window.open\ (URL) = Fore.URL(F_{url}) \\ >0\, Window.open(URL) \neq Fore.URL(F_{url}) \end{cases} \tag{13}$$

iv. **Abnormal URL Shortening:** Phishers use URL shorteners to obfuscate phishing URLs when requesting unsuspecting users to log-in their accounts through a link especially on social networking sites. If a link shortening time stamp pattern and the Number of encoders is not similar to genuine URL Shortening Service (USS) such as Bitly, goo.gl, Owl.ly, Deck.ly, Su.pr etc. then it is likely to be phishing. Otherwise, it is legitimate.

$$\boldsymbol{F2}_4 = \begin{cases} 0\, \boldsymbol{Link_{t,enc(n)}} = \boldsymbol{USS_{t,enc(n)}} \\ >0\, \boldsymbol{Link_{t,enc(n)}} \neq \boldsymbol{USS_{t,enc(n)}} \end{cases} \tag{14}$$

v. **Downloadable malicious code:** Most phishing sites or emails contain an instruction to download certain files which are used to perpetrate crimeware-based attacks. If a webpage contains an active download link which contains specified extension such as .aaa, .abc, .exx, .help_restore,

Table 4
Frequency rate of selected web features.

S/N	Web feature	Frequency rate
1	Domain name check	0.65
2	Domain name in the URL path	0.31
3	Server Form Check/Pop-up	0.28
4	Abnormal URL shortening	0.24
5	Downloadable malicious code	0.73
6	Abnormal cookie domain	0.23
7	Age of domain	0.42
8	Port number behaviour	0.28
9	SSL certificate	0.58
10	Blacklist domain	0.77

Please cite this article as: A. A. Orunsolu, A. S. Sodiya and A. T. Akinwale, A predictive model for phishing detection, Journal of King Saud University – Computer and Information Sciences, https://doi.org/10.1016/j.jksuci.2019.12.005

6–7 length extension of random characters, then it is phishing and suspicious. Otherwise, it is legitimate.

$$F2_5 = \begin{cases} 0\, Down(link) \neq malicious\ extension \\ >\ 0\, Down(link) = malicious\ extension \end{cases} \quad (15)$$

Algorithm 2. Feature Assessment Frequency Analysis

Input: Data size, *p;* original feature set, *n;* threshold, θ, class, C

Output: Subset of top features of dimension *m(fs)*

Begin

For $i = 1$ *to* n do begin
For $j = 1$ *to* p do begin
Activate Mendeley app
$a = 1$
Selects$_i \in H_P$
Compute CFS (s) using {
$s(f_i, f_2, \cdots f_n, c)$as input
For $i = 1$ *to* n do begin
Initialize appropriate correlation factor, t
r = calculate _ correlation (f, c)
If $(t > r)$ **then**
Append $f_i \in s_n$ into m
End
order m in descending value of t
Remove f with lower rank
Return predominant f as *f(s)*
End}
If $(f_i(s) > \theta)$ **then** add to *m(fs)*
$c = c + 1$
If $(a \leq n)$ *goto* 4
End for
End for
Return feature set *m(fs)*

End

3.4. The behaviour of a webpage (F3-Filter)

The behaviour of a webpage describes the features of a webpage which are related to how the webpage handles its underground processes. Such underground processes may include how the transmission is made between the HTTP cookies and web server, the WHOIS history, port number type, certificate type e.g. self-issued or trusted third-party etc. The following features are considered for classification by FSM after performing frequency assessment using algorithm 2.

i. **Abnormal Cookie domain**: This feature checks how the transmission of text data is done by a web server to a web client. Information about client machine/users is usually maintained in this text data, which is sometimes called HTTP cookies. If a website has a domain cookie ($\boldsymbol{D_C}$) which is in a foreign domain, then it may be deceptive as most benign websites have their domain cookies or no cookies ($\boldsymbol{D_{url}}$).

$$\boldsymbol{F3}_1 = \begin{cases} 0\ \ \boldsymbol{D_C} = \boldsymbol{D_{url}} \\ >0\ \ \boldsymbol{D_C} \neq \boldsymbol{D_{url}} \end{cases} \quad (17)$$

ii. **Age of domain:** This feature checks the age of the domain name of a particular URL or the URL extracted from the action attribute of a form using the WHOIS API search. Many phishing pages claim the identity of a known brand which has a relatively long history. If the age of the domain does not correspond to its WHOIS lookups, then it is likely to be deceptive.

$$\boldsymbol{F3}_2 = \begin{cases} 0\ \boldsymbol{D_{url\_age}} = \boldsymbol{WHOISAPI_{url.age}} \\ >0\ \ \boldsymbol{D_{url\_age}} \neq \boldsymbol{WHOISAPI_{url.age}} \end{cases} \quad (18)$$

iii. **Port Number behaviour:** This feature compares the port number part of a domain name with the stated protocol part of a URL. If the protocol does not match the port number, then the page is a phishing site.

$$\boldsymbol{F3}_3 = \begin{cases} 0\ \boldsymbol{D_{url.portNumber}} = \boldsymbol{HTTPS_{url.portNumber}} \\ >0\ \boldsymbol{D_{url.portNumber}} \neq \boldsymbol{HTTPS_{url.portNumber}} \end{cases} \quad (19)$$

iv. **SSL Certificate:** The Secure Socket Layer certificate is often used every time-sensitive information is being transferred by a user to an honest website. This certificate can either be self-signed by a website or offered by a trusted third party such as GeoTrust, VeriSign etc. There is a high level of probity with a trusted third-party certificate than a self-signed certificate. So, checking that the SSL certificate is offered by the trusted issuer is a feature of the legitimate website. Otherwise, it is suspicious or phishy.

$$\boldsymbol{F3}_4 = \begin{cases} 0\ \boldsymbol{Sensitive_{url.}} = \boldsymbol{Protocol_{ssl}} \\ >0\ \boldsymbol{Sensitive_{url.}} \neq \boldsymbol{Protocol_{ssl}} \end{cases} \quad (20)$$

v. **SSL Certificate:** The Secure Socket Layer certificate is often used every time-sensitive information is being transferred by a user to an honest website. This certificate can either be self-signed by a website or offered by a trusted third party such as GeoTrust, VeriSign etc. There is a high level of probity with a trusted third-party certificate than a self-signed certificate. So, checking that the SSL certificate is offered by the trusted issuer is a feature of the legitimate website. Otherwise, it is suspicious or phishy.

$$\boldsymbol{F3}_4 = \begin{cases} 0\ \boldsymbol{Sensitive_{url.}} = \boldsymbol{Protocol_{ssl}} \\ >0\ \boldsymbol{Sensitive_{url.}} \neq \boldsymbol{Protocol_{ssl}} \end{cases} \quad (20)$$

vi. **Blacklist domain:** Since blacklisting of suspicious URLs has produced promising results in phishing detection, the blacklist domain is used as a feature in our approach to managing a list of locally detected phishing sites to bypassed superfluous computation on an already known malicious domain. This feature provides the advantages of providing resources for updating our feature corpus and reduce overheads. If the domain name used in the action field of a login form or URI ($\boldsymbol{D_C}$) is found in the blacklist domain, then it is likely to be phishing. Otherwise, it is legitimate.

$$F3_5 = \begin{cases} 0\ \ D_C \not\subset blacklist\ domain \\ >0\ \ D_C \subseteq blacklist\ domain \end{cases} \quad (21)$$

3.5. Classifier construction for the proposed system

Given an identity (i.e. malicious or legitimate) and a set of features, the task of determining the genuineness of a transaction is executed by a classification algorithm. A classification algorithm automatically learns how to make accurate predictions of unknown instances based on the past or trained observations. The accuracy (i.e. True Positives, False positives, Recall Rate, True Negatives

Please cite this article as: A. A. Orunsolu, A. S. Sodiya and A. T. Akinwale, A predictive model for phishing detection, Journal of King Saud University – Computer and Information Sciences, https://doi.org/10.1016/j.jksuci.2019.12.005

etc.) and the resources-requirements (i.e. training time, response time, memory overhead, etc.) with which the predictions are made determine the efficiency of such classification algorithms.

In the problem of phishing detection, the use of NB and SVM are common on varieties of the feature vector with different volumes of data dimensionality (Moghimi and Varjani, 2016). Frequency analysis of different classifiers from extant literature indicates high adoption of these two classifiers, especially in phishing problem definition due to their simplicity and high accuracy (Anwar et al., 2017; Dhanalakshmi and Chellappan, 2013). Motivated by the past investigations of NB and SVM on phishing dataset, our classification method employs these two classifiers on the same feature set to evaluate their performance. The basic assumptions in the construction of these two classifiers are to investigate the sensitivity of the proposed feature set on these classifiers.

3.6. The Naïve Bayes classifier

The Naïve Bayes is a simple and effective text classification algorithm which use the joint probabilities of words and categories to estimate the probabilities of categories given a document (Anwar et al. 2015). The conditional independence assumption can be formally expressed as:

$$P(A|C=c)=\prod_{i=1}^{n}P(A_i|C=c) \tag{22}$$

where each attribute set or feature set A = $\{A_1, A_2, \cdots, A_n\}$ consists of n attribute values. With the conditional independence assumption, instead of computing the class conditional probability for every grouping of A, only estimate the conditional probability of each A_i, given C (Isa et al. 2008). This makes the Naïve Bayes approach more practical because it does not require a very large training set to obtain a good estimate of probability (Abu et al., 2011). Besides, the classifier can easily handle missing attribute values by omitting the probability when calculating the likelihoods of membership in each class. To classify a test sample, the NB classifier computes the posterior probability for each class C as:

$$P(C|A)=\frac{P(C)\prod_{i=1}^{n}P(A_i|C)}{P(A)} \tag{23}$$

Eq. (23) indicates that by observing the value of a particular feature, A_i, the prior probability of a particular category, Ci, P(Ci) can be converted to the posterior probability, P ($Ci|A_i$), which represents the probability of a particular feature, A_i being a particular category, Ci. Algorithm 3 shows how NB classifier works based on the constructed model for phishing classification.

Algorithm 3: Naïve Bayes Classifier (NBC)

Input: Feature vector, *Fv*; training dataset, *D*,

Output: *L*, the Phishing classification result

Begin:

1. Initialize each F ∈ Fvextracted from W
2. **While** Fv ∈ W≠0**do** {
3. Calculate the prior probability P(C_k) for each class C_k
4. Calculate the Conditional probability of P (A_{ik} | Ci) for each F∈ Fv
5. Classify each training example, e,∈ D with maximum posterior probabilities
6. Update the probabilities of each e∈ D based on the probability of classification
7. Classify new e← L}
8. **End while**
9. **End**

3.7. The support vector machine classifier

Support Vector Machine is a powerful learning method that has been successfully applied to text categorization. The main objective of SVM as a binary classification algorithm is to find a hyperplane that separates data points in the space of possible inputs (Diale et al., 2016). Assuming there is a given set of linear separable training samples $(x_i, y_j)_1 \leq_i \leq N$, $x_i \in R^d$, and that $y_i \in \{-1,1\}$ is the class label which $\boldsymbol{x_i}$ belongs to (Zhang et al., 2001). The general form of linear classification function is:

$$g(\mathbf{x})=\mathbf{w}\mathbf{x}+b \tag{24}$$

which corresponds to a separating hyperplane $\mathbf{w}\cdot\mathbf{x}$ + b = 0. In SVM principle, g($\boldsymbol{x}$) is normalized to satisfy | g (**x**) | $\geq 1 \forall \mathbf{x}_i$, so that the distance from the closest point to the hyperplane is expressed as:

$$\frac{1}{\|\boldsymbol{w}\|} \tag{25}$$

There are many separating hyperplanes, the one for which the distance to the closest point is maximal is called *optimal separating hyperplane* (OSH): the hyperplane lying half-way in between the maximal margin. As the distance to the closest point in the hyperplane is given in (25), the solution **w** has an expansion expressed as:

$$\mathbf{w}=\sum_i \alpha_i y_i \mathbf{x}_i \tag{26}$$

Subject to:

$$\text{yi}(\mathbf{w}\cdot\mathbf{x}i+b)\geq 1, i=1,\cdots,N \tag{27}$$

If (α_i,. . .,α_N) denotes the N non-negative Lagrange multipliers associated with constraints in (27), then a uniquely OSH can be constructed by solving the constrained quadratic programming problem (Qian et al., 2007; Zhang et al., 2001). The classification function can, therefore, be formalized as:

$$F(\mathbf{x})=sign\left(\sum_i \alpha_i\, y_i x\ +b\right) \tag{28}$$

Algorithm 4 shows how SVM classifier works based on the constructed model for phishing classification.

Algorithm 4: Support Vector Machine Classifier

Input: *Fv*, Feature vector from Feature extractor; *D*, training Data;

Output: *L1*- (Phishing), *L0*- (Non-Phishing)

Begin

1. Mark $F \in F_v \forall D$ into two classes, c_{1-2}: phishing I^+ and non-phishing I^o
2. While $\exists c_{1-2} \forall F \in F_v$ do
3. {Construct the training data (x_i, *yi)*
   a. For each $x_i \in I^+ \cup I^o$
   b. While $y_i = \begin{cases} +1, if\ x_i \in I^+ \\ -1, if\ x_i \in I^o \end{cases}$
4. Construct classification function$f(x)=\sum_i \alpha_i y_i k(x_i, x)+b$
   a.Calculate the score for each *F*.
   Score(F_i) = f(x_i)
   b.Sort all *F* by score and return new result}
5. End While
6. ***End***

Please cite this article as: A. A. Orunsolu, A. S. Sodiya and A. T. Akinwale, A predictive model for phishing detection, Journal of King Saud University – Computer and Information Sciences, https://doi.org/10.1016/j.jksuci.2019.12.005

3.8. Incremental construction of the proposed system

The components of the predictive model for phishing detection are organized into a system using the incremental construction of the component-based systems (Lau et al., 2012). This approach has the advantage of providing practical solutions for managing scale and complexity in system development. In the proposed system, there are two kinds of components for building the system incrementally: atomic and composite.

The atomic component consists of a computation unit and an invocation connector. In this case, each filter (e.g. URL filter) is regarded as an atomic component of the system. The computation unit described the function or methods or expressions evaluated on the data in the system while the invocation connector is used to communicate with the other atomic or composite components. The computation unit of the atomic unit performed computations when invoked within its scope without calling other computation units. That is, the computation unit encapsulates computations for which the heuristics in each atomic unit are defined.

On the other hand, the composite component is constructed as a resultant computation unit from atomic components using composition connectors. The composition connectors represent controls which trigger computation for coordinating component. The composition connectors use several controls such as sequencing, pipe etc. for achieving efficient system construction. Using composition connector by arity, Fig. 2 presents the incremental construction model for the proposed scheme. Algorithm 5 depicts the workflow within the incremental construction model.

Algorithm 5: Incremental construction algorithm

Input: *Suspicious URL*

Output: *URL status*

Begin

1. **If** *url* ∈ *e*F1(*url*) // *e*F1*(url)* means URL features
2. Compute total value of *e*F1*(url)*
3. Generate feature vector for *eF1(url)*
4. Use invocation connector to communicate *e*F1*(url)* to the next atomic subcomponent
5. Return ***False***
6. **Else**
7. **If** *url* ∈ *e*F2(*url*) // *e*F2(*url*) means webpage properties
8. Compute total value of *e*F2(*url*)
9. Generate feature vector for *e*F2(*url*)
10. Use invocation connector to communicate resultant feature vector (*eF1(url)* + *e*F2(*url*)) to next atomic subcomponent
11. Return ***False***
12. **Else**
13. **If** *url* ∈ *e*F3(*url*) // *e*F3(*url*) means webpage behaviour
14. Compute total value of *e*F3(*url*)
15. Generate feature vector for *e*F3(*url*)
16. Use invocation connector to communicate resultant feature vector (*eF1(url)* + *e*F2(*url*) + *e*F3(*url*)) to composite component
17. Return ***False***
18. **Endif**
19. Normalize and vectorize resultant feature vector
20. Train the classifier with the resultant feature vector
21. Generate predictive model for the classifier
22. Test the predictive model

End

4. Implementation and evaluation

The implementation procedure involves the use of JSoup HTML parser and Waikato Environment for Knowledge Analysis (Weka). The JSoup HTML parser is used to extract the feature set from the DOM of the webpage. Besides, a Java library called Secure Socket Extension is used to extract third party information relating to a particular domain during the feature extraction process. This provides an effective way to examine all the relevant features and tags from the parsed page to examine their status.

On the other hand, the WEKA application is used to develop the classification model for the extracted features through a feature vector generation process. These features are built from data corpus consisting of legitimate and phishing pages before being transported into WEKA as CSV file format. The extracted features are stored as CSV file which is then read by the SVM and NB classifier model in WEKA for prediction process. Unlike most classification approaches which are usually single machine learning-based, the classification model for the proposed method is a two-level classifier to determine the performance of the selected features in the different machine learning model.

4.1. Evaluation parameters and experimental analysis

The performance of the proposed system is evaluated by using five standards parameters consisting of True Positive Rate, False Positive Rate, True Negative Rate, False Negative Rate and Accuracy. These are the standard performance metrics to evaluate any phishing detection system. Let P denotes the total number of phishing sites and L represents the total number of legitimate sites. Using the following notations:

P.a as phishing sites classified as phishing
P.b as phishing classified as legitimate
L.a as legitimate classified as legitimate
L.b as legitimate classified as phishing
Then, the following definitions follow:

(i) True Positive Rate: this is the rate of phishing websites that are classified as phishing out of the aggregate phishing websites.

$$TPR = \frac{P.a}{P} \times 100 \tag{29}$$

(ii) False Positive Rate: this is the rate of phishing websites that are classified as legitimate out of the aggregate phishing websites

$$FPR = \frac{P.b}{P} \times 100 \tag{30}$$

(iii) False Negative Rate: this is the rate of legitimate websites classified as phishing out of the aggregate legitimate websites

$$FNR = \frac{L.b}{L} \times 100 \tag{31}$$

(iv) True Negative Rate: this is the rate of legitimate websites classified as legitimate out of the aggregate legitimate websites

Please cite this article as: A. A. Orunsolu, A. S. Sodiya and A. T. Akinwale, A predictive model for phishing detection, Journal of King Saud University – Computer and Information Sciences, https://doi.org/10.1016/j.jksuci.2019.12.005

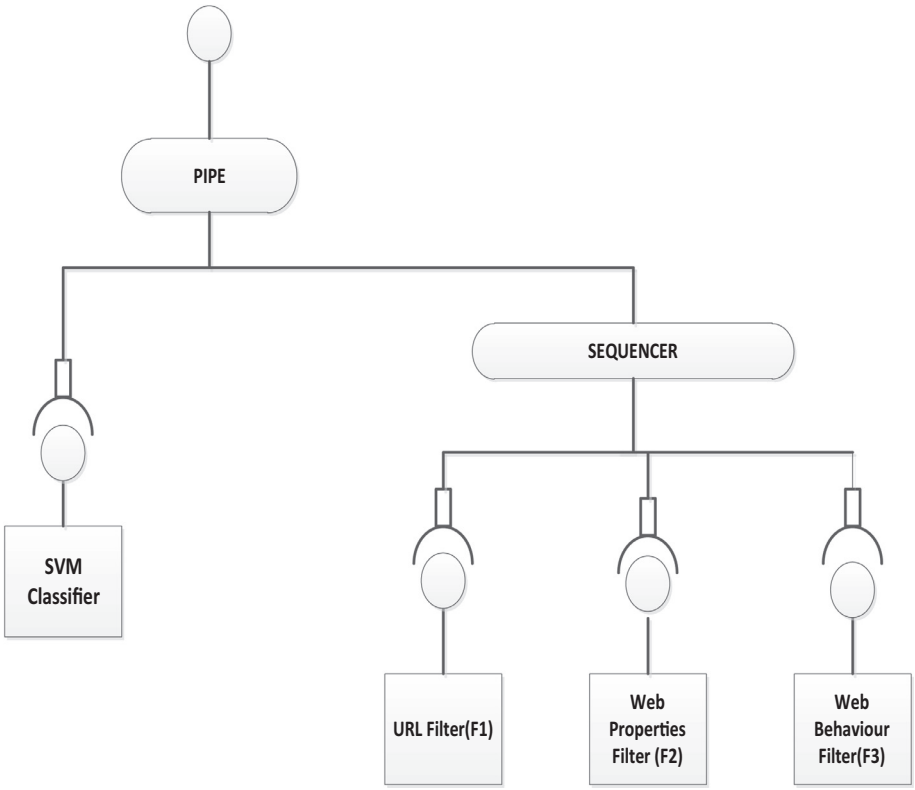

Fig. 2. The Incremental Construction Model.

$$TNR = \frac{L.a}{L} \times 100 \tag{32}$$

(v) Accuracy: this is the rate of phishing and legitimate websites which are identified correctly to all the websites

$$A = \frac{P.a + L.a}{L + P} \times 100 \tag{33}$$

In addition, Mathew's Correlation Coefficient (MCC) is calculated to determine the quality of the prediction model. When MCC approach unity, it is indicative that the system achieves near to perfect prediction and therefore, it is a reliable detection system for phishing. The following equation represents the MCC

$$MCC = \frac{TPR \times TNR - FPR \times FNR}{\sqrt{(TPR + FPR)(TPR + FNR)(TNR + FPR)(TNR + FNR)}} \tag{34}$$

4.2. Description of experimental dataset and experiments

Two publicly available datasets consisting of phishing URLs and benign URLs are used to evaluate the performance of the proposed phishing detection architecture. The experimental datasets are obtained from data corpus consisting of 40,000 unique phishing URLs from PhishTank dataset (2018) and 1,000,000,000 legitimate URLs from Alexa. PhishTank is a community-based security service that contains a database of verified and labelled phishing URLs reported by a community of human volunteers. On the other hand, Alexa provides commercial web traffic data, global ranking and analytics of top-most visited URLs. These two data sources are chosen as they contain verified datasets used in most anti-phishing research for benchmarking evaluation results.

As the life of phishing sites are often short-lived, we crawled only 2541 phishing URLs from the available phishing dataset for the experimental purpose. In the same vein, we selected 2500 clean URLs from Alexa for the same purpose. These sites were randomly selected from the entire research data corpus for experimentation and evaluation purposes. Each entry of the experimented data corpus is unique to avoid repetitive processing/evaluation of the same webpage more than one time. Thus, the experimentation and evaluation process are non-data redundancy processes and the corpus consists of popular commonly targeted sites in phishing attacks. These sites are easily targeted because unsuspecting users are glibly deceived by the look and feel of any phishing page mimicking such genuine pages. A simple homomorphic attack using similar character can make this attack to be possible. These sites include well-known e-commerce sites, financial houses, social networking sites, government agencies etc. These selected web pages are used to verify the performance of the prediction model.

4.3. Experimental Environment and evaluation tool

Experiments were conducted and subsequently evaluated on a computer system with an Intel Core i5 processor with 4 GB RAM and 500 GB HDD. A number of experiments were conducted to evaluate the performance and accuracy of the proposed anti-phishing scheme. The evaluation procedure of the prediction model is based on WEKA 3.6.13 with the default setting of Test Options on appropriate fold-type. A 10-fold cross-validation experiment was selected before the evaluation process was activated on the test dataset. This involves randomly splitting of test dataset into ten equal sub-samples, from the ten sub-samples a single sub-sample is used for the final validation of the model while the remaining other sub-samples are used to train the system. Hence, the proposed predictive model was built on 90% of the dataset and validated on the remaining 10%. This process was repeated 10

Please cite this article as: A. A. Orunsolu, A. S. Sodiya and A. T. Akinwale, A predictive model for phishing detection, Journal of King Saud University – Computer and Information Sciences, https://doi.org/10.1016/j.jksuci.2019.12.005

times and after the validation, a single estimation is computed. This estimation is the average of the ten iterations.

The incentive to apply cross-validation experiment is to fit the performance of a model outside the training dataset. Other reasons are:

i. To verify the error performance of the system's predictive model. In this case, the errors associated with the SVM and NB predictive model in phishing detection based on the feature vector.
ii. To validate the training data set via validating each subset. This is to ensure more confidence in the model trained on the training set.
iii. To assess how the evaluation results of the predictive model will generalize to an independent data corpus on phishing/ benign sites.

The cross-validation experiments involve the evaluation of the predictive model for True Positive rate and False Negative rate using the phishing dataset and the evaluation of True Negative rate and False Positive rate using the legitimate dataset. The datasets for the experiment consist of non-overlapping legitimate and phishing websites preprocessing and read in into the predictive model for evaluation.

4.4. Experiment 1: evaluation of the subset of feature set

In this experimental approach, we obtained the performance of the proposed approach in term of TPR and FPR concerning the subset of the feature set. This was done by building the predictive model to consider only the subset of the selected features. Specifically, the subset of the feature set was selected from the three feature categories used in this work. These features were selected to measure their impact on the evaluation process and determine if there is any contributory influence or complementing effects on the other features that are not considered. The features selected for this experiment are nine (9) consisting of IP features, long URLs (URL category), domain name check, server form check, abnormal URL shortening, (Web behavioural features), abnormal cookies domain, age of domain, port number behavior and SSL certificate (Web properties features).

The experimental dataset consists of 1353 instances of phishing URLs that are randomly selected from the subset of the initial experimental dataset. Our experiment on phishing dataset indicates that SVM returned 86.77% TPR and 13.22% FPR while NB gave 84.70% TPR and 15.29% FPR using 10-fold cross-validation (Fig. 3). Figs. 4 and 5 showed a Receiver Operating Characteristics curve which examines the change in FPR to the change in TPR of the proposed classifiers i.e. $ROC = \frac{\Delta y}{\Delta x}$ where Δy *is the change in False Positive Rate and* Δx *is the change in True Positive Rate*. The objective of ROC estimation is to investigate the increase in FPR with the increase in TPR while changing the discrimination threshold of the predictive model. The Area Under Curves (AUC) values for both predictive models are promising as shown in the figures. The FSM extraction technique is used to extract the phishing features before the proposed model is applied to build the classifier. These results indicated that the subset of the feature set play a significant role in the prediction process as their performance is well over average. However, further, improvement is still required for effective phishing prediction in a critical scenario like e-banking where high FPR is a limiting factor.

4.5. Experiment 2: evaluation of the entire feature set

Unlike the first experiment, this experiment 2 accessed how effective the adoption of the entire feature set in terms of all the evaluation metrics. These features are communicated to the two classifiers used in this approach through the invocation connector of the respective feature category in the incremental construction algorithm. Both the phishing and legitimate datasets are used in the evaluation process for this experiment. The evaluation process was conducted using a 10-fold cross-validation experiment.

	True Results	
	Phishing URLs (1353)	
Prediction	SVM	NB
Classified as phishing	TPR =1174	TPR =1146
Classified as legitimate	FNR =179	FNR =207

Fig. 3. Statistics for experiment 1.

The experimental dataset for experiment 2 consists of 2541 phishing instances and 2500 legitimate instances. The evaluation statistics for the second experiment indicates a TPR of 99.96, FNR of 0.04, TNR of 99.96, FPR of 0.04, an accuracy of 99.96 and MCC of 0.9996 for both classifiers. Fig. 6 shows the statistics of the experimental results. The results suggested that those other features (i.e. @ symbol in the URL path, hexadecimal in the URL path, domain name in the URL path, downloadable malicious code and blacklist domain) have profound contributory influence or complementing effects on the overall prediction accuracy. This implies that the effectiveness of the first nine (9) features has been improved by the contributory influence of the other features. For instance, although the length of a URL may be elongated by phishers employing @ symbol in the URL path, the omission of such feature may be bypassed by a phisher using shorter URL while engaging the use of the symbol or IP address.

These evaluation results show that the proposed predictive model will generalize correctly on both phishing and benign dataset to detect phishing attacks with insignificant false ratio and high accuracy rate. The MCC is approximately equal to 1 (i.e. 0.9996) which indicates that the predictive power of the machine learning model in the cross-validation experiment approaches near perfection. These evaluation results indicate how effective the integration of various "historical" features can be in the fight against phishing attacks. These figures have shown that the proposed approach has classification accuracies ranging from 84% (incomplete feature set) to 99.96% (complete feature set) with an average of 99.15% ROC-AUC (Fig. 7). Fig. 8 indicated the visualization of the feature extractor plot for the legitimate URLs with only one misclassification indicated with a circle. For more information on the evaluation datasets, see references on Kaggle.

4.6. The efficiency of the proposed anti-phishing model

The efficiency of the anti-phishing scheme is usually computed using three evaluation parameters namely Precision, Recall and F1-Score. The precision, p, is used to measure the rate of phishing instances which are identified correctly as the instances detected as phishing. That is, p is the number of correct positive results divided by the number of all positive results returned by a classifier. The Recall, r, is a measure of phishing instances identified correctly as existing phishing instances. This implies that r is the

Please cite this article as: A. A. Orunsolu, A. S. Sodiya and A. T. Akinwale, A predictive model for phishing detection, Journal of King Saud University – Computer and Information Sciences, https://doi.org/10.1016/j.jksuci.2019.12.005

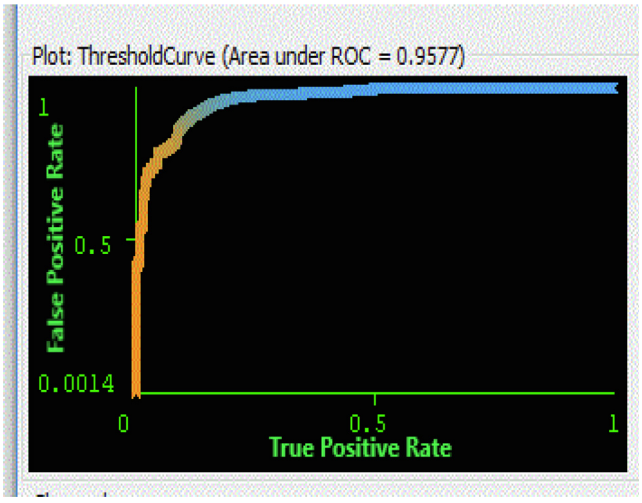

Fig. 4. ROC-AUC of the Naïve-Bayes Model.

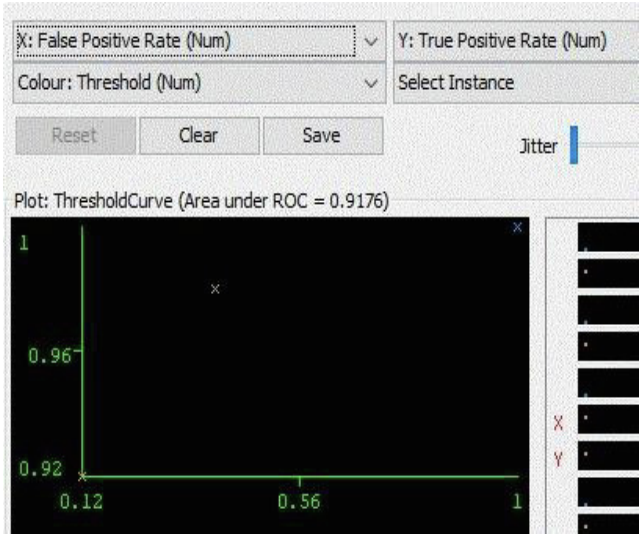

Fig. 5. ROC-AUC of the SVM model.

True Results

Prediction	Phishing URLs (2541)	Legitimate URLs (2500)
Classified as phishing (2541)	TPR =2540	FPR =1
Classified as legitimate (2500)	FNR =1	TNR =2499
Total	2541	2500

Fig. 6. Statistics of the experimental results.

number of correct positive results divided by the number of all relevant samples (all samples that should have been identified as positive). The F1-Score is the harmonic mean of Precision and Recall and F1-score reaches its best value at 1 and worst at 0. The basic mathematical representation of these parameters is given as:

$$\text{i. } p = \frac{TP}{TP + FP} \tag{35}$$

$$\text{ii. } r = \frac{TP}{TP + FN} \tag{36}$$

$$\text{iii. } F_1 Score = \frac{2.p \times r}{p + r} \tag{37}$$

The computed values for these three parameters from the 10-fold cross-validation experiment is compared with other approaches (Table 5). These results indicate that the proposed model has high efficiency when applying for the phishing detection process. The similarity in the values of the three metrics (i.e. precision, recall, F1-score) for our approach stems from the results of the TPR, FPR and FNR of the proposed predictive model. The efficiency of the proposed system is also compared Phishing Detection

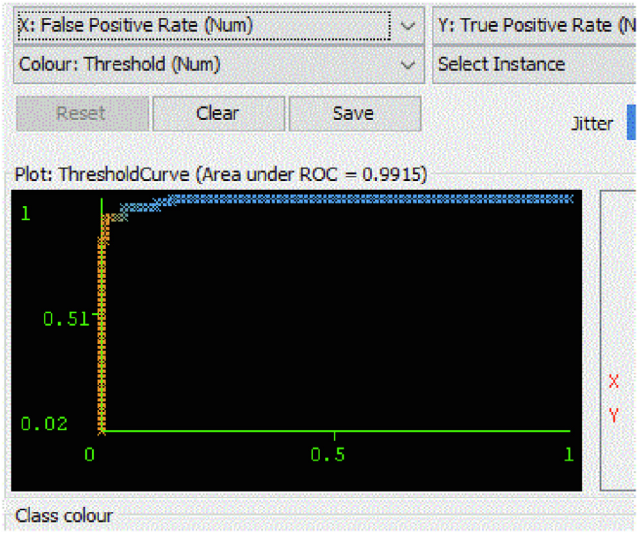

Fig. 7. Average ROC-AUC of the proposed model.

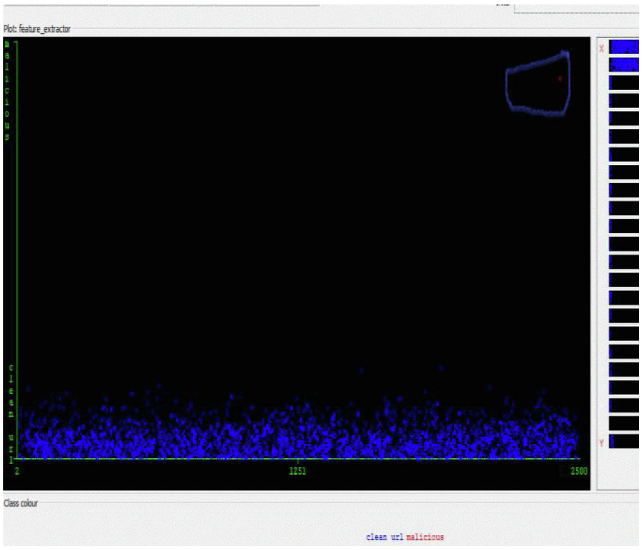

Fig. 8. Visualization of feature extractor plot.

Please cite this article as: A. A. Orunsolu, A. S. Sodiya and A. T. Akinwale, A predictive model for phishing detection, Journal of King Saud University – Computer and Information Sciences, https://doi.org/10.1016/j.jksuci.2019.12.005

using Multi-Filter Approach (PhiDMA) (Sonowal and Kuppusamy, 2017) and Phishing page detection via classifiers from page layout feature (PPDCPLF) (Mao et al. 2019). The specificity of the approach which estimates the measure of correctly identified legitimate URLs is defined as:

$$S = \frac{TP}{TN + FP} \tag{38}$$

Thus, the specificity of the proposed approach indicated 0.9969 in contrast to the PhiDMA which produced 0.9054

The PhiDMA model incorporates five layers namely Auto upgrade whitelist Layer, URL feature layer, Lexical signature layer, string matching layer and accessibility score comparison layer to detect phishing pages. On the other hand, the PPDCPLF uses page layout similarity to detect phishing pages.

4.7. Runtime analysis of the predictive model

The average runtime, av_p, performance is evaluated to show the overhead in terms of memory usage and time duration for the predictive model to complete the detection process. The system used for the runtime analysis is a corei3 processor speed of 7th generation running Windows OS. Using a standardized timing procedure, the average runtime is estimated as the time taken in loading the testing dataset in either .csv or. JSON or other supported extension/format to the time taken for the WEKA application to generate the evaluation statistics for the test. This time includes the time taken for normalizing, t_n, and preprocessing, t_p, of the dataset once the predictive model is built. The building process involves the extraction of the feature vector into the parameter settings of the chosen classifier algorithm in the WEKA application, t_e. Thus, the average runtime performance, $av_p = \sum_{i=1}^{3} t_i/n$.

The runtime for all the experiments is presented in Table 6. Although both predictive models achieve the same results, the runtime performance of NB is better than the SVM. Experiment analysis indicates that SVM uses an average of 1580 ms while the NB uses an average of 10 ms. Thus, the runtime analysis indicates the worst case of an average of 1580 ms and the best case of an average of 10 ms for the proposed model based on the extracted feature vector to conclude the status of a suspicious webpage.

Table 7 presents the runtime analysis of the proposed method with other approaches with known data. The comparison is based on the worst case of 1580 ms of the proposed SVM model.

4.8. Deployment option for the predictive model

The efficiency of the anti-phishing scheme can be affected by the deployment options used in the real-life implementation of the system. For instance, most client-side deployments suffer from the use of a specific browser to secure online communication against phishing attacks (e.g. SpoofGuard is deployed on Mozilla). In addition, intensive administration (e.g. undue updates of a browser for a security patch, installation, etc.) and JavaScript exploits limited the efficiency of client-side deployment (Aparna and Muniasamy, 2015; Han et al. 2012). Similarly, the use of anti-phishing scheme as a server-side filter is being challenged by trust and third-party involvement. Also, (Varshney et al., 2016a) posited in their review that the high-dimensional cost and training datasets constituted problems for ML-based APS as a browser add-on or a lightweight IDS. Motivated by these challenges, we proposed to deploy our anti-phishing techniques as a middleware.

Middleware technology is one of the viable alternatives to the challenges of client/server anti-phishing deployment (Ofuonye and Miller, 2013; Gupta et al. 2016). The primary advantages of middleware deployment are that it leverages the solution as a service model thereby making the proposed scheme accessible and executable by a large number of users, ensure that the anti-phishing service remains efficient by automatically adding new filters and transparency to both client and server. For instance, Gowtham and Krishnamurthi, 2014 implemented the anti-phishing scheme as a web service thereby offering the advantage of high upgradability with high accuracy. Although the use of middleware deployment option may generate the problem of scalability on the cyberspace, the increasing adoption of cloud computing infrastructure makes it easy to leverage this limitation. Based on these facts, we argue that using middleware deployment option for our predictive model will guarantee high usability, security and timely upgrade.

Table 5
Comparison of the proposed approach with other methods based on efficiency.

S/N	APPROACH	Precision	Recall	F1-Score
1	PhiDMA	91.23	90.55	90.88
2	PPDCPLF	92.70	92.05	92.10
3	Our method	99.96	99.96	99.96

Table 6
Runtime Performance of the experiments.

S/N	Experiment	Runtime (ms)
1	SVM model phishing dataset (cross-validation)	2280
2	NB model phishing dataset (cross-validation)	10
3	SVM model legitimate dataset (cross-validation)	880
4	NB model legitimate dataset (cross-validation)	10

Table 7
Comparison of Runtime Analysis with other techniques.

S/N	WORK	RUNTIME ANALYSIS
1	Gowtham et al. (2017)	29,333 ms
2	Kaur and Kalra (2016)	21,412 ms
3	The Proposed Method	1,580 ms

4.9. System evaluation and results analysis

The approach proposed in this research work addresses the limitations of other anti-phishing techniques in terms of certain parameter measurements such as computational efficiency, robusticity and use of upgrade already discussed in section 2. In this section, performance assessment on key evaluation parameters such as True Positive rate, False Positive rate and Accuracy is compared with other existing techniques. This comparison is based on the most recent works in anti-phishing techniques. Table 8 presents the performance statistics that indicate that the proposed method is better than the existing anti-phishing model in all these parameters evaluation. Fig. 9 presents the graphical illustrations in terms of detection accuracy of the proposed methods in comparison with other methods.

4.10. Discussion and limitation

The predictive model implemented in this paper leverages the sufficient integration of the subset of top relevant phishing "fingerprints", which presents the advantages of detecting phishing based on available high-ranking feature vectors already discussed in extant literature (Aburrous et al. 2010). Experimental results indicated that the approach achieved significant detection accuracy when compared with other related works. These promising results are clear inventive that phishing can still be adequately tackled with proper selection of high-ranking feature vector. Unlike most existing anti-phishing techniques which are implemented as a browser plugin, the implementation of this method is considered

Please cite this article as: A. A. Orunsolu, A. S. Sodiya and A. T. Akinwale, A predictive model for phishing detection, Journal of King Saud University – Computer and Information Sciences, https://doi.org/10.1016/j.jksuci.2019.12.005

Table 8
Comparison of related works with the proposed method.

S/N	Work	Phishing data	Benign data	Total	TPR	FPR	Accuracy
1	Sonowal and Kuppusamy (2017)	667	995	1662	90.54	5.82	92.72
2	Kaur and Kalra (2016)	1078	846	1924	99.44	0.56	97.59
3	Hota et al. (2018)	4150	4116	8266	99.19	0.81	99.27
4	Gowtham et al. (2017)	2129	1546	3675	99.53	0.45	99.54
5	Chin (2018)	3718	1185	4903	96.90	0.03	98.39
6	Tan et al. (2017)	500	500	1000	99.20	7.80	91.51
7	The Proposed Method	2541	2500	5041	99.96	0.04	99.96

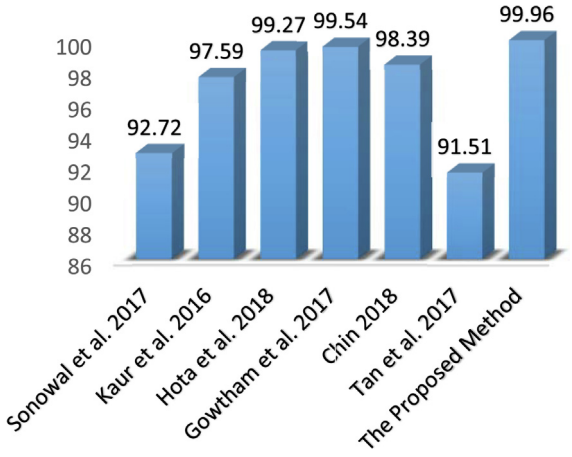

Fig. 9. Accuracy assessment of the proposed scheme with other methods.

to be middleware-based. Hence, some inherent problems such as superfluous computations of machine learning models are prevented on the client or server-side which may slow down the system during uptime.

A number of limitations are identified and discussed in the foregoing paragraph. For instance, if a phisher mimics a webpage using sophisticated tools, then the look and feel of such website may be a replica of the legitimate page. In this case, the use of visual attributes may not produce significant positive results. However, since our approach involves attributes that examines the web contents, such replica may not reduce the detection accuracy of our approach. Although the use of visual attributes may assist when image-based webpages hosting phishing attacks are launched, which shows that the non-inclusion of such attributes is a minus to our work. This obvious limitation may be subdued since the use of such a strategy is not very predominant in phishing attacks (Varshney et al., 2016a).

The abuse of search engine strategy is another technique which may affect the efficiency of the proposed method. In this attack, false positives are associated with newly hosted benign URLs or hijacked legitimate URLs running phishing activities. Such phishing activities sometimes involve the installation of malicious code on users' system. As a result of this, our approach avoids total dependence on search engine strategy and adopted a more practical approach of extracting more robust features from the URL, web document properties and web behaviour attributes. Although, this comes with some overheads on the system resources as search engine strategy has been adjudged to be lighter than most ML-based techniques (Gupta et al. 2016; Jain and Gupta (2017)).

Another attack scenario common to anti-phishing techniques is pharming attack, which exploits vulnerabilities that allows phishing URLs to share the same domain name with benign URLs. Although our approach does not consider the use of forward and reverse DNS query validation, which may solve the problem, such techniques come with additional burden on the DNS server in resolving the queries (Varshney et al., 2016a). As such, the use of such features in our future work will require a great deal of fine-tuning techniques to reduce its inherent computational flaws.

From our evaluation dataset, we observed that most of the test domains are the English language based and as such, we cannot adequately determine the language independence of our approach in mitigating phishing attack. This may limit the performance of our approach, especially where the large percentage of the testing dataset is non-English.

The last limitation of our approach relates to the absence of target identification in phishing attacks. Since the goal

of the anti-phishing system is to effectively detect phishing activities, the target identification may seem like a secondary goal of APS which is desirable but negotiable.

Although the use of domain name attributes in our approach can effectively provide some succour in this regard, full integration of such methods will enhance the performance of our predictive model, especially in alert notification to the real website to take down the offensive counterpart.

5. Conclusions

The main idea in this article is to investigate how existing phishing feature datasets can be sufficiently integrated into an effective countermeasure. To achieve this, the ranking of different feature categories from extant literature was used to select our proposed feature vector.

Based on this premise, the proposed approach extracted some features from the URL, webpage properties and webpage behaviour using frequency assessment analysis. In the end, an incremental construction model was used to organize the features and its associated machine learning models into a system for flexibility and manageability. The approach was evaluated using experiments on the classifiers consisting of NB and SVM with a dataset consisting of 2541 phishing pages and 25,000 legitimate pages. The results indicate an agreeable runtime of less 2,000 ms and evaluation metrics consisting of 99.96 True Positives, 99.96 True Negatives, 0.04 False Positive and 0.04 False Negatives. From these results, the proposed approach presents a superior anti-phishing scheme when compared with other existing approaches under the given experimental circumstances.

In the future, our further research study will consider the following: (i) investigating the application of our approach on proprietary middleware such as SOAP (ii) exploring the use of our design as mobile apps for smartphone-based phishing attacks (iii) investigating the appropriateness of our design in the emerging IoT-based phishing attacks (iv) investigating the contributory influence or complementing effects of various features using more intensive study on appropriate theoretical framework.

6. Author agreement

We agreed with the guidelines and rules of publication related to the Journal of King Saud University.

Please cite this article as: A. A. Orunsolu, A. S. Sodiya and A. T. Akinwale, A predictive model for phishing detection, Journal of King Saud University – Computer and Information Sciences, https://doi.org/10.1016/j.jksuci.2019.12.005

7. Compliance with ethical standards

a. funding: this study was not funded by any agency
b. conflict of interest: Orunsolu a.a declares that he has no conflict of interest. Sodiya a.s declares that he has no conflict of interest. akinwale a.t declares that he has no conflict of interest.
c. ethical approval: this article does not contain any studies with human participants or animals performed by any of the authors

References

Aburrous, M., Hossain, M. A., Thabatah, F. and Dahal, K. 2008. Intelligent Phishing Website Detection System using Fuzzy Techniques.
Aburrous, M., Hossain, M., Dahal, K., Thabtah, F., 2010. Experimental case studies for investigating e-banking phishing techniques and attack strategies. Cognit. Comput.
Abu Afza A., Farid D., Rahman C. 2011. A Hybrid Classifier using Boosting, Clustering and Naïve Bayesian Classifier. WCSIT.
Adebowale M., Lwin K., Sanchez E and Hossain M. 2018. Intelligent Web-Phishing Detection and Protection Scheme using integrated Features of Images, Frames and Text. Expert System with Applications.
Anwar, T., Abu-Kresha, M., Bakry, A., 2017. An efficient method for web page classification based on text. International. J. Eng. Comput. Sci.
Anti-Phishing Working Group (APWG) 2017 security report.
Aparna S., Muniasamy K. 2015. Phish Indicator: An Indication for Phishing Sites. Artificial Intelligence and Evolutionary Algorithms in Engineering Systems.
Barraclough, P., Sexton, G., 2015. In: Phishing website detection fuzzy system modelling. IEEE, London, UK, pp. 1384–1386.
Bottazzi, G., Casalicchio, E., Cingolani, D., Marturana, F., Piu, M 2015. MP-Shield: A Framework for Phishing Detection in Mobile Devices. In: IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM), pp. 1977-1983. IEEE.
Chandrashekar, G., Sahin, F., 2014. A Survey on Feature selection methods. Comput. Electric. Eng. J. Springer. 40, 16–28.
Chin, T., 2018. PhishLimiter: A Phishing Detection and Mitigation Approach using Software-Defined Networking. IEEE.
CSO Online report on phishing activities. Accessed 2016 (http://www.csoonline.com/articles)
Chiew, L., Chang, H., Sze, N., Tiong, K., 2015. Utilization of website logo for phishing detection. Comput. Secur. J.
Diale M., Walt C., Celik T., Abiodun M. 2016. Feature Selection and Support Vector Machine Hyper-parameter Optimization for Spam Detection. In proc. PRASA-RonMech. South Africa.
Dhamija, R., Tygar, J.D., Hearst, M., 2006. Why phishing works. Proc. of. In: Factors in Computing Systems. ACM Press, pp. 581–590.
Dhanalakshmi, R., Chellappan, C., 2013. Detecting Malicious URLs in E-mails- An Implementation. In Proceedings of Conference on Intelligent Systems and Controls. AASRI Proceia 4, 125–131.
Dunlop M, Groat S, Shelly D., 2010. GoldPhish: using images for content-based phishing analysis. In: International conference on internet monitoring and protection. Barcelona, Spain, pp. 123–128.
Gowtham, R., Gupta, J., Gamya, P.G., 2017. Identification of phishing web pages and their target domains by analyzing the feign relationship. J. Informat. Secur. Appl. 35, 75–84.
Gowtham, R., Krishnamurthi, I., 2014. PhishTackle-a web services architecture for anti-phishing. Cluster Compt.
Gupta S., Singhal A., Kapoor A., 2016. A Literature Survey on Social Engineering Attacks: Phishing Attack. International Conference on Computing, Communication and Automation.
Gupta, B., Tewari, A., Jain, K., Agrawal, P., 2017. Fighting against phishing attacks: state of the art and future challenges. Neural Comput. Appl. https://doi.org/10.1007/s00521-016-2275-y.
Han W, Cao Y, Bertino E and Yong J. 2012.Using automated individual white-list to protect web digital identities. Expert Systems with Applications.
Hamid, A., Abawajy, J., 2014. An approach to profiling phishing activities. Journal of computer and security. Elsevier Press.
Hota, H.S., Shrivas, A.K., Hota, Rahul, 2018. An ensemble model for detecting phishing attack with proposed remove-replace feature selection technique. Procedia Comput. Sci. 132, 900–907. https://doi.org/10.1016/j.procs.2018.05.103.
Isa, D., Lee, L., Kallimani, V., Rajkumar, R., 2008. Text document pre-processing using bayes formula for classification based on the vector space model. Comput. Informat. Sci. J.
Jain, A., Gupta, B., 2017. Two-level authentication approach to protect from phishing attacks in real-time. J. Ambient Intell Human Comp. https://doi.org/10.1007/s12652-017-0616-z.
Jain, A.K., Gupta, B.B., 2016. A novel approach to protect against phishing attacks at client side using auto-updated white-list. EURASIP J. Inf. Secur. 2016, 1–11.
Lau K., Ng K., Rana T., Tran M. 2012. Incremental Construction of Component-based Systems. In Proceeding of CBSE Bertinoro Italy.
Kaggle: https//www.kaggle.com/softline/machine-learning-model-to-identify-malicious-url.
Li, Y., Yang, Z., Chen, X., Huan, H., Liu, W., 2019. A stacking model using URL and HTML features for phishing webpage detection. Fut. Generat. Comput. Syst.
Kaur, D., Kalra, S., 2016. Five-tier barrier anti-phishing scheme using a hybrid approach. Inform. Secur. J. A Global Perspect.
Khonji M, Iraqi Y, Jones A. 2013. Phishing Detection: A Literature Survey. IEEE.
Mao, J., Bian, J., Tian, W., Zhu, S., Wei, T., Li, A., Liang, Z., 2019. Phishing Page detection via classifier from page layout feature. EURASIP J. Wireless Commun. Network. 43.
Moghimi, M., Varjani, A.Y., 2016. New rule-based phishing detection method. J. Exp. Syst. Appl. 53, 231–242.
Mohammed A., Furkan A., and Sonia C. 2015. Why phishing still works: User strategies for combating phishing attacks. International Journal of Human-Computer Studies. Volume 82. pp. 70-82. Elsevier Press
Ofuonye, E., Miller, J., 2013. Securing web-clients with instrumented code and dynamic runtime monitoring. J. Syst. Softw.
Orunsolu A, Sodiya A, Akinwale A, Olajuwon B.(2017) An Anti-Phishing Kit Scheme for Secure Web Transactions. In the proceedings of 3rd ICISSP Conference, Porto Portugal, SCITEPRESS.
Orunsolu, A., Afolabi, O., Sodiya, S., Akinwale, A., 2018. A Users' Awareness Study and the Influence of Socio-Demography Perception of Anti-Phishing Security Tips. Acta Informatica Pragensia. J. Univers. Econ. Czech Republic 7 (2), 138–151.
Phishtank dataset (2018). http://www.phishtank.com.
Qabajeh, I., Thabtah, F., Chiclana, F., 2018. A recent review of conventional vs. automated cybersecurity anti-phishing techniques. Comput. Sci. Rev.
Qian, T., Vai, M., Wavelet, Xu.Y., 2007. Anal. Appl.
Sonowal, G., Kuppusamy, K.S., 2017. PhiDMA- A phishing detection model with a multi-filter approach. J. King Saud Univers.
Sonowal G., Kuppusamy K.S. 2019. MMSPhiD: A Phoneme based Phishing Verification Model for Persons with Visual Impairments. Information and Computer Security Journal. Emerald Publishing Limited.
Shabtai, A., Kanonov, U., Elovici, Y., Glezer, C., Weiss, Y., 2012. "Andromaly": a behavioural malware detection framework for android devices. J. Intellig. Inform. Syst. 38 (1), 161–190.
Stats and Trend 2017 Security Report.
Tan C., Chiew L., Sze N. 2017. Phishing Webpage Detection Using Weighted URL Tokens for Identity Keywords Retrieval. Lecture Notes in Electrical Engineering. Vol. 398.
Toolan F., Carthy J. 2010. Feature selection for Spam and Phishing detection. In: Proceedings of the eCrime Researchers Summit (eCrime), Dallas, Texas, USA, pp. 1–12.
Varshney, G., Misra, M., Atrey, P., 2016a. A survey and classification of web phishing detection Schemes. Secur. Comm. Networks.
Varshney, G., Misra, M., Atrey, K., 2016b. A phish detector using lightweight search features. Comput. Secur. 62, 213–228.
Wikipedia, 2018.
Zhang L., Lin F., Zhang B. 2001. Support Vector Machine Learning for Image Retrieval. IEEE.
Zouina, M., Outtaj, B., 2017. A novel lightweight URL phishing detection system using SVM and similarity index. Human-Cent. Comput. Informat. Sci. J.

Please cite this article as: A. A. Orunsolu, A. S. Sodiya and A. T. Akinwale, A predictive model for phishing detection, Journal of King Saud University – Computer and Information Sciences, https://doi.org/10.1016/j.jksuci.2019.12.005