Jurnal
Teknologi

Full Paper

HYBRID FEATURES - BASED PREDICTION FOR NOVEL PHISH WEBSITES

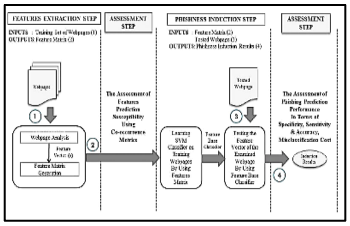
Hiba Zuhair[a][b], Mazleena Salleh[b]*, Ali Selamat[b][c]

[a]Al-Nahrain University, Baghdad, Iraq
[b]Faculty of Computing, Universiti Teknologi Malaysia, 81310, UTM Johor Bahru, Johor, Malaysia
[c]Center of Communication and Information Technologies (CICT), Universiti Teknologi Malaysia, 81310, UTM Johor Bahru, Johor, Malaysia

Article history
Received
26 November 2015
Received in revised form
14 January 2016
Accepted
10 October 2016

*Corresponding author
mazleena@utm.my

Graphical abstract


FEATURES EXTRACTION STEP
INPUTS : Training Set of Webpages (1)
OUTPUTS: Feature Matrix (2)
ASSESSMENT STEP
PHISHNESS INDUCTION STEP
INPUTS : Feature Matrix (2)
Tested Webpage (3)
OUTPUTS: Phishness Induction Results (4)
ASSESSMENT STEP
Webpages
1
Webpage Analysis
Feature Vector (s)
Feature Matrix Generation
2
The Assessment of Features Prediction Susceptibility Using Co-occurrence Metrics
Learning SVM Classifier on Training Webpages By Using Features Matrix
Feature Base Classifier
Tested Webpage
3
Testing the Feature Vector of the Examined Webpage By Using Feature Base Classifier
4
Induction Results
The Assessment of Phishing Prediction Performance In Terms of Specificity, Sensitivity & Accuracy, Misclassification Cost

Abstract

Phishers frequently craft novel deceptions on their websites and circumvent existing anti-phishing techniques for insecure intrusions, users' digital identity theft, and then illegal profits. This raises the needs to incorporate new features for detecting novel phish websites and optimizing the existing anti-phishing techniques. In this light, 58 new hybrid features were proposed in this paper and their prediction susceptibilities were evaluated by using feature co-occurrence criterion and a baseline machine learning algorithm. Empirical test and analysis showed the significant outcomes of the proposed features on detection performance. As a result, the most influential features are identified, and new insights are offered for further detection improvement.

Keyword: Hybrid features, novel phish websites, prediction susceptibility, co-occurrence criterion, phishness induction

© 2016 Penerbit UTM Press. All rights reserved

1.0 INTRODUCTION

In the last decade, the web data flow has shown a rapid expansion of phish websites that practically lure targeting users to acquire their own sensitive information by masquerading them as a trustworthy entity in the web environment. Phish website is a form of phishing, where phishers imitate a legitimate website by exploiting specific deceptions and innovative social tactics for digital identity theft and then monetary gains [1-3]. To intuitively tackle phish websites, researchers have introduced numerous anti-phishing techniques and involved various detection methods such as non-classification and classification based methods those assisted by different features [1-4]. In the literature of anti-phishing, the classification based methods outperformed the others due to the use of machine learning and data mining algorithms. However, there is still high false detection errors causing inaccurate detection against novel phish website. Phishers often evolve novel phish websites exploiting new and more sophisticated deceptions to bypass the aforesaid detection methods for more insecure intrusions, identity theft and illegal profits [4, 5]. Hence, the steady escalating of novel phish websites becomes the most intricate issue needs to be considered seriously [3, 4, 5]. For the problem at hand, this paper has made the following contributions: (i) this paper proposes new, hybrid and predictive features for detecting those novel phish websites, (ii) an experimental strategy has been conducted by using an optimized assessment criterion in order to identify features' prediction susceptibility and analyse features' influence on detection performance. More precisely, the proposed features encompasses a hybridity of web page content and web page URL features.

Furthermore, they leveraged the most dynamic attributes that phishers could use to impersonate their targeting legitimate websites such as those of embedded objects, cross site scripting, and those crafted in URLs hosting non-English webpages. It is hoped that contributions has been made by this paper would help researchers in the field of phishing mitigation with a great knowledge and understanding about the causality between the features proposed herewith, their prediction potentials, and the overall detection performance on novel phish websites as well as those prevalent ones.

The remaining sections of this paper are as follows: Section 2 presents a background of phishing and anti-phishing, commonly used features along with related works. Section 3 states the developed methodology that pursued in this work. Then, Section 4 presents the dedicated experimental strategy, the experimental dataset, and the results along with a deep discussion and comparative analysis. Also, it synthesizes several implications for future work. Section 5 outlooks on the proposed features and developed classification methodology, draws conclusions.

1.1 The State-of-The-Art

1.1.1 Phishing and Anti-Phishing

It is well known that phishing is a form of online fraud to acquire the Internet users' confidential and personal information through the identification theft for financial gains. Typically, the phishers accomplish this fraudulent process in a strategic manner with the aid of a spoofed email or fake website along with a social engineering tactic [6]. According [7] a phish website is any web page that impersonates illegally a trustworthy website and it is crafted on the behalf of a third party (phisher) who intends to confuse users and acquire their credentials for fraud activity and illegal profits. Radical escalating of phish websites on the Web causes direct losses (e.g. financial loss and identity theft) to the users and enterprises and indirect losses (e,g. impacts on customer's trust in an online service, or e-banking transaction, or reputation of a financial organization and brand). To this end, many researchers have applied different anti-phishing techniques to combat phishing activities and mitigate their damages particularly over the last few years. Researchers' developments varied in their deployments including detection methods, features and information sources. To give an insight on such detective methods, they were categorized into non-classification and classification based methods, and each category was decomposed into sub-categories such as blacklist and whitelist methods, machine learning classifiers and data mining methods, hybrid, and information-flow methods [8, 9, 10]. Among all the aforesaid categories and sub-categories, the classification methods outperform their competitors due to the integration of features and machine learning classifiers or data mining rules to build effective phishing classification models. Therefore, researchers continually develop their achievements with the aid of classification methods by optimizing their inductive bias and extending the features in use towards obtaining a holistic characterization on phishing deceptions with minimal detection errors.

1.1.2 Phishing Deceptive Features

In the light of classification methods, prior researchers have deployed different types of features for phishing detection on websites. Deployed features were varied from static to dynamic features due to their nature and the source of extraction. Static features can be extracted from the webpage source and URL without full execution of the web page itself. Whereas, dynamic features can be retrieved during the webpage source rendering and execution [8, 9, 10].

Consequently, the deployed features have made different discriminative contributions at predicting phishing susceptibility. Some feature might being non-informative and irrelevant to phishing class, and then they might negatively influence the overall effectiveness of classification method [11]. That is the most intricate issue encountered with the utilization of classification based methods. On the other hand, phishers often advance their deceptions and exploit more innovative ones to bypass the existing anti-phishing campaigns. Example of such evolutionary features are those of specific embedded and dynamic objects, host files, domain names and top level domains, and cross site scripting codes. Phishers craft such kinds of features on websites to inject malicious codes, harvest passwords, and redirect the visiting users to fake websites. Moreover, advancement of phishers' deceptions involved exploiting URLs of trustworthy websites hosting with different natural languages rather than English [2, 11-13]. As such, they are being able to evade anti-phishing techniques relied on textual features exclusively. As a result, the aforesaid evolutionary types of features yield misclassification costs, and then more potential security risks and monetary losses. Day after day, evolutionary features become the key challenge versus effective anti-phishing techniques assisted by machine learning methods [11, 14, 15]. Such challenge is emerged as the most salient research agenda which demands persistent exploration of new deployments to promote existing generations of features. Beyond this, it is essential to find out highly discriminative features for accurate classification on phish and non-phish web sites. Examples of evolutionary features that have been deployed in the literature are categorized and displayed in Figure 1. Also, the relevant issues of each feature category is depicted briefly in Table 1.

1.1.3 Related Work

Classification based detection methods were developed for intuitive filtering and protection against phish websites, therefore, they were heavily applied to the client-side level in between the interaction of users with web environment. In this context, Table 2 enlists examples of such client side filters along with their relative merits and demerits.

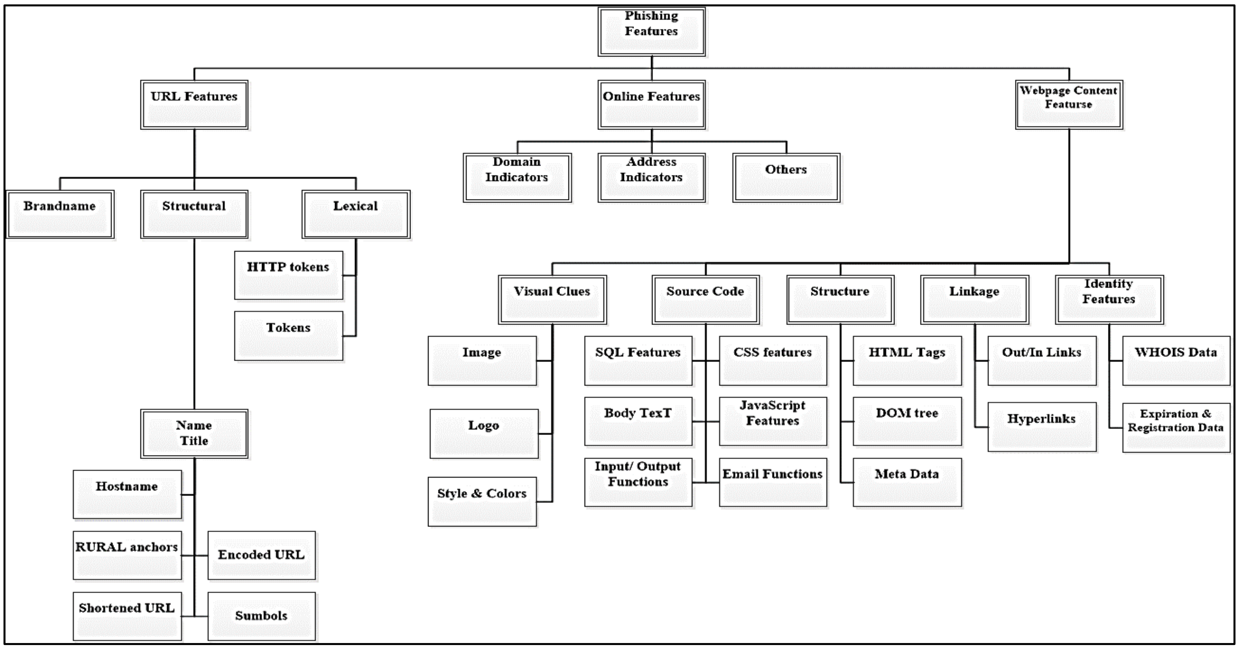


Figure 1 Categories of phishing features as they adopted in [3, 6-9, 11, 16-32]

Table 1 Comparison of different feature categories

Feature Category	**Advantage**	**Disadvantage**
Webpage Content features	Comprehensiveness & widely usage	Challenge of obfuscation, code coverage, malicious code injection and delivery
URL features	Easy extraction & widely usage	Challenge of phish detection with high sensitivity,
Online features	Easy extraction	Limited usage and requirement of external resources
Hybrid features	High comprehensiveness	Complex extraction process and requirements

Table 2 Characterization of related work with respect to their leveraging of novel features

Related Work	**Brief Description**	**XSS Features**	**Embedded Objects Features**	**Language Independent Featur**
[6]	It classifies phishing emails and webpages using classifier and Google's blacklist	No	No	No
[16]	Utilize DOM tree objects, HTTP transactions and some webpage components to detect phishes	No	No	Yes
[17]	It Submits fake credentials before and after actual user's credentials.	Yes	No	Yes
[18]	It identifies phishing websites by using Bayesian filter and DOM tree.	No	No	Yes
[19]	It sends bogus credentials when a webpage is detected as phishing to avoid information leakage.	No	No	Yes
[9, 20]	Identifies phishing websites by using FSM and several features.	Yes	No	Yes
[21]	It maintains blacklist of phishing URLs using TLD and DNs features	No	No	Yes
[22]	It is based on both lookup and a SVM classifier that checks features derived from websites URL, text and linkage**.**	No	No	No
[23]	Extract features of webpage identity and compare them with the current domain using search engine.	No	No	No
[25]	It utilizes recorded legitimate URLs in a whitelist and Bayesian algorithm to verify URL's legitimacy	Yes	No	Yes

Regarding Table 2, a hybrid based anti-phishing tool, Google Toolbar, was presented to incorporate an upgraded Google phishing blacklist with a machine learning classifier [7]. In [16], phishing filtering was devoted using textual and DOM objects features alongside Support Vector machine classifier (SVM). Two key components are involved in filtering, they are information retrieval algorithm to extract textual features and Chi-squared criterion to select the most discriminative ones. Another example is PhishGuard [17], it was asserted to keep track users' submissions and their bogus credentials during login activities. Meanwhile, B-APT [18], was adopted to filter websites of US financial institutions exclusively. It relied on Bayesian filter and a whitelist of examining tokens and objects to identify phishing websites. Later, Bogus Bitter was developed by [19] to deliver a plenty of bogus credentials along with those actual ones as a way to keep track phish websites. Whereas, PhishTester was developed to mitigate cross site scripting phish attacks exclusively by exploiting vulnerabilities in web browsers and detecting suspicious codes through the flow of information [9, 20]. In contrast, PhishNet was attained in [21] as an upgraded blacklist with a classifier and new URL features in order to proof-check whether an examined URL useful for resolving DNS lookup. Then, an evolutionary hybrid anti-phishing tool (PhishBlock) was developed and introduced in [22] for classifying the phishness of URL and textual features extracted from examined websites. Meanwhile, CANTINA$^+$ was proposed by some researchers working at Carnegie Mellon University as an extensible CANTINA with extra discriminative features. CANTINA$^+$ was developed to involve an information flow mechanism along with machine learning classifiers for better filtering on novel phishing attacks [23]. Whereas, AIWL was an automated individual whitelist built in browsers to protect users during online transaction [24].

Overall the aforesaid phishing filters, have several outstanding issues in leveraging evolving phish websites particularly those crafted with cross site scripting (XSS) features, embedded objects based features, and newly exploited URLs of non-English hosted webpages (Table 2). Therefore, further efforts should be dedicated to explore new features with high prediction susceptibility for holistic characterization and accurate classification on novel phishes. This, in turn, will improve the effectiveness of existing anti-phishing techniques with minimal computations, misclassification cost, false alarms, and performance overhead.

2.0 METHODOLOGY

This section articulates the solution to the problems at hand through four steps, website analysis, features extraction, features assessment and phishing classification.

2.1 Website Analysis

To extract a hybrid set of features from web page content and URL, the relevant components and parts in the web page source code were parsed, and a Document Object Model (DOM) tree was constructed as the node-based representation of the examined web page. In fact, Document Object Model (DOM) is a standard platform of the World Wide Web Consortium (W3C) that allows dynamical access to the webpage's content, structure and style by programs and scripts [34]. DOM tree includes nodes denoting different components of the webpage that they could be rendered as a rooted tree with nodes, and each node represents a constituent tag [34]. By this way, all components, elements, attributes and textual parts are represented by the leaf nodes of DOM tree.

The webpage source code represents the starting node or the root of DOM tree, and then the DOM tree continues to extend its branches to the lowest level, where all the leaf nodes exist (Figure 2). Then, the constructed DOM tree is treated as a graph *G (V, E)* in which any two vertices are connected by exactly one path in order to extract the wanted features [33-35]. The set of vertices in the graph represents the set of nodes in the DOM tree signifying the set of tags in the webpage source code.

To extract URL features, the webpage URL was tokenized into terms including lexical features, tokens, specific irregularities and structural elements. To retrieve the required features during the tokenization process, the classic TF-IDF metric was used as per Equation (1) [25]:

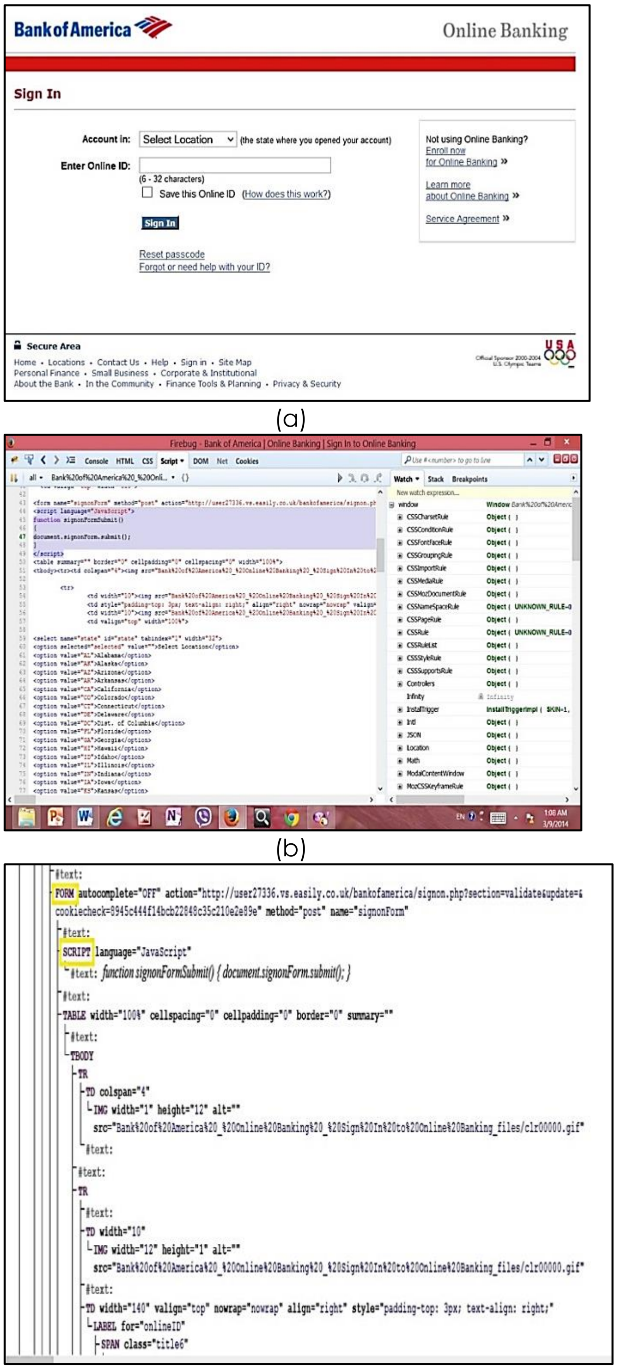
$$tf - idf(w) = tf(w) \cdot idf(w) \qquad (1)$$

Where *term frequency tf(w)* indicates the count of occurrences of *w* in the webpage, *inverse document frequency idf(w)* represents the general importance of *w* in the whole collection and *w* is the term in query.

Then, a feature vector was generated so that each webpageV_j, a vector of multiple features $V_j = \{V_{j,1}, \dots, V_{j.|V_j|}\}$ where$|V_j|$, is the total number of extracted features from that webpage. Each feature $V_{j,i}$ returned either a binary value or a numeric value. The binary values are computed as the union of their corresponding feature's values, while the numeric features in their values are combined by taking the average value of the corresponding features. Finally, the obtained feature vectors were combined into a multi-dimensional vector (Feature Matrix) to include all the extracted feature vectors corresponding to all web pages in the training set. Also, the corresponding webpage class label (phish or non-phish) was set in the first column of each row in the feature matrix.

2.2 Features Extraction

Novel deceptive features may rely highly on some embedded components and scripts in the webpage' source code as well as lexical and structural token in the webpage URL address [11, 14, 20, 23]. Thus, these features are categorized based on the parts from where they can be extracted as explained below:



Bank of America Online Banking

Sign In

Account in: Select Location (the state where you opened your account)

Enter Online ID:

(6 - 32 characters)

□ Save this Online ID (How does this work?)

Sign In

Reset passcode

Forgot or need help with your ID?

Not using Online Banking?

Enroll now for Online Banking »

Learn more about Online Banking »

Service Agreement »

Secure Area

Home • Locations • Contact Us • Help • Sign in • Site Map

Personal Finance • Small Business • Corporate & Institutional

About the Bank • In the Community • Finance Tools & Planning • Privacy & Security

Official Sponsor 2000-2004 U.S. Olympic Teams

(a)

Firebug - Bank of America | Online Banking | Sign In to Online Banking

```
<form name="signonForm" method="post" action="http://user27336.vs.easily.co.uk/bankofamerica/signon.ph
<script language="JavaScript">
function signonFormSubmit()
{
document.signonForm.submit();
}
</script>
<table summary="" border="0" cellpadding="0" cellspacing="0" width="100%">
<tbody><tr><td colspan="4"><img src="Bank%20of%20America%20_%20Online%20Banking%20_%20Sign%20In%20to%2

        <tr>
                <td width="10"><img src="Bank%20of%20America%20_%20Online%20Banking%20_%20Sign%20In%20
                <td style="padding-top: 3px; text-align: right;" align="right" nowrap="nowrap" valign=
                <td width="10"><img src="Bank%20of%20America%20_%20Online%20Banking%20_%20Sign%20In%20
                <td valign="top" width="100%">

<select name="state" id="state" tabindex="1" width="32">
<option selected="selected" value="">Select Location</option>
<option value="AL">Alabama</option>
<option value="AK">Alaska</option>
<option value="AZ">Arizona</option>
<option value="AR">Arkansas</option>
<option value="CA">California</option>
<option value="CO">Colorado</option>
<option value="CT">Connecticut</option>
<option value="DE">Delaware</option>
<option value="DC">Dist. of Columbia</option>
<option value="FL">Florida</option>
<option value="GA">Georgia</option>
<option value="HI">Hawaii</option>
<option value="ID">Idaho</option>
<option value="IL">Illinois</option>
<option value="IN">Indiana</option>
<option value="IA">Iowa</option>
<option value="KS">Kansas</option>
```

```
window                Window Bank%20of%20Americ
  CSSCharsetRule      Object { }
  CSSConditionRule    Object { }
  CSSFontFaceRule     Object { }
  CSSGroupingRule     Object { }
  CSSImportRule       Object { }
  CSSMediaRule        Object { }
  CSSMozDocumentRule  Object { }
  CSSNameSpaceRule    Object { UNKNOWN_RULE=0
  CSSPageRule         Object { }
  CSSRule             Object { UNKNOWN_RULE=0
  CSSRuleList         Object { }
  CSSStyleRule        Object { }
  CSSSupportsRule     Object { }
  Controllers         Object { }
  Infinity            Infinity
  InstallTrigger      InstallTriggerImpl { SKIN=1,
  Intl                Object { }
  JSON                Object { }
  Location            Object { }
  Math                Object { }
  ModalContentWindow  Object { }
  MozCSSKeyframeRule  Object { }
```

(b)

```
#text:
FORM autocomplete="OFF" action="http://user27336.vs.easily.co.uk/bankofamerica/signon.php?section=validate&update=&
cookiecheck=8945c444f14bcb22848c35c210e2e89e" method="post" name="signonForm"
 #text:
 SCRIPT language="JavaScript"
  #text: function signonFormSubmit() { document.signonForm.submit(); }
 #text:
 TABLE width="100%" cellspacing="0" cellpadding="0" border="0" summary=""
  #text:
  TBODY
   TR
    TD colspan="4"
     IMG width="1" height="12" alt=""
     src="Bank%20of%20America%20_%20Online%20Banking%20_%20Sign%20In%20to%20Online%20Banking_files/clr00000.gif"
    #text:
   #text:
   TR
    #text:
    TD width="10"
     IMG width="12" height="1" alt=""
     src="Bank%20of%20America%20_%20Online%20Banking%20_%20Sign%20In%20to%20Online%20Banking_files/clr00000.gif"
    #text:
    TD width="140" valign="top" nowrap="nowrap" align="right" style="padding-top: 3px; text-align: right;"
     LABEL for="onlineID"
      SPAN class="title6"
```

(c)

Figure 2 Example of webpage presentation (a) the brov view, (b) HTML view and (c) view of related DOM tree

A. Embedded objects and features: specific components with their objects and attributes exist inside their relative HTML parts and tags. They can be exploited by phishers for their powerful functionality to imitate the webpage content, insert and hide forged content and external links for redirection to fake websites. Furthermore, phishers make these objects invisible to avoid phish detection mechanisms that leverage traditional webpage content features rather than these features (Table 3) [25, 26].

B. Cross site scripting (XSS) features: they are exploited by phishers as a type of deceptions that enables them to hide, inject and deliver doubtful and malicious scripts to the client side by using client side scripting languages. Further, they are exploited to defeat existing anti-phishing techniques aiming at severe security risks because they can be interpreted by web browsers [9, 20, 27]. Basically, most of these features can be extracted from script tags involving code pieces, calls and their events, and the native JavaScript functions, etc. (Table 3).

C. Language independent features: phishers usually use certain features and modification on URLs to host phish websites that imitate legitimate ones. Regarding the literature, it is very common for potential phishing URLs to contain terms, irregularities, and indicators that have been used for estimating phishing susceptibility in websites [28-31] as presented in Table 3.

Table 3 Characterization of proposed features in terms of their relevant categories, sub-categories and extraction sources

Feature	Feature Category (S)	Feature Subcategory (S)	Extraction Source
Embedded objects based features	Webpage Content	Linkage, Structural, and Source Code	DOM Components, HTML Tags, Body Text, I/O functions, Hyperlinks, In/Out Links
XSS-based features	Webpage Content	Source code	JavaScript cod
Language independent features	URL & Online Features	Structural and Lexical features	IP address, Hostname, RURAL anchors, Domain & Address Indicators

2.3 Features Assessment

To assess the prediction susceptibility of the proposed features, *co-occurrence criterion* was computed. The co-occurrence calculation relies on an optimized equation of that adopted in [16]. Assume that $\forall f \in F$ and $\forall\, d \in D$. $C_{f,d}$ is set as the number of occurrences of *f* in d and $C_{f,D}$ is set as the number of occurrence of each *f* in *D*, where *D* is the set of all the examined instances and *F* is the set of features belonging to each instance d in *D*. Then, the co-occurrence calculation is defined as per Equation (2) [32]:

$$C_{f,D} = \sum_{d \in D} C_{f,d} \qquad (2)$$

In the case of our study, novel phishing features could belong to valid phish and not valid phish (suspicious) web pages rather than phish web pages exclusively. Hence, we re-present the aforesaid criterion in terms of valid phish, (*D*) and not valid phish (*D*'). then, the *co-occurrence value* of each extracted feature *f* is computed as follows:

$$C_f = \frac{C_{f,D} - C_{f,D'}}{C_{f,D} + C_{f,D'}} \qquad (3)$$

Where $D, D', and\ f$ are the occurrence of each feature *f* belongs to feature vector *F* in all instances that included in *D* and *D*'. $C_{f,\,D}$ and $C_{f,\,D'}$ denote the co-occurrence values of feature *f* with respect to all instances in *D*, and the occurrence of feature *f* with respect to all instances in *D*'. Then, C_f is the accumulative co-occurrence of $C_{f,\,D}$ and $C_{f,\,D'}$.

2.4 Phishing Classification

SVM classifier is the most commonly used machine learning classifier to obtain the optimal separating hyper plane between two classes [35-37]. It guarantees the lowest level of error rate because of its generalization ability and handling of high dimensional feature space. SVM classifier produces two output classes [35-37] represented by two labels (+1) and (-1) to induct the class of a given feature vector whether it is phishing or not phishing.

To do so, *V* is a feature matrix denotes all the webpages in the learning dataset such that $V = \{V_1 \quad V_j \quad V_{|V|}\}$ and V_j is the feature vector of each webpage as $V_j = \{v_{j,1} \quad v_{j,i} \quad v_{j,|V_j|}\}$, where $|V|$ and $|V_j|$ are the number of feature vectors and features in each feature vector, respectively. Then, $v_{j,i}$ is the value of each i^{th} feature of j^{th} feature vector V_j, where $0 \le v_{j,i} \le 1$, $i = 1, 2, 3, \dots, |V_j|$ and $j = 1, 2, 3, \dots, |V|$, given that $V = \{V_j\}_{j=1}^{|V|}$ is a set of $|V|$ training feature vectors or alternatively the M-dimensional feature matrix. Each V_j is labelled by $y_j \in \{1, -1\}$ with $y_j = 1$ and $y_j = -1$ which indicates the membership of V_j in the (class 1) and (class 2) as per Equation (4) [36, 37].

$$f(x) = \sum_j \alpha_j y_j K(V', V_j) + b \qquad (4)$$

Where α_i and b are obtained by a quadratic algorithm, *V*' is the unlabeled webpage and V_j is the feature vector of a training webpage. The function $K(V', V_j)$ maps the space of input webpage to higher dimensions where training webpages in the dataset are learned individually. As such, the classifier assessed the relevance of the input feature established an inductive function *Y* = *f*(*V*, *y*) to induct its class as either phish or legitimate.

After applying the induction function on all feature vectors included *F* during the training task, a *feature base classifier* was obtained for further classification on a given tested web page during the testing task. Features were extracted from the incoming webpage and being represented as a feature vector V_{new}. Then, it was learnt with *feature base classifier* to produce its class label V'_{new} as either phish or legitimate [35-37]. Typically, an illustration of the overall classification scenario which adopted from [35-37] is presented in Figure 3.

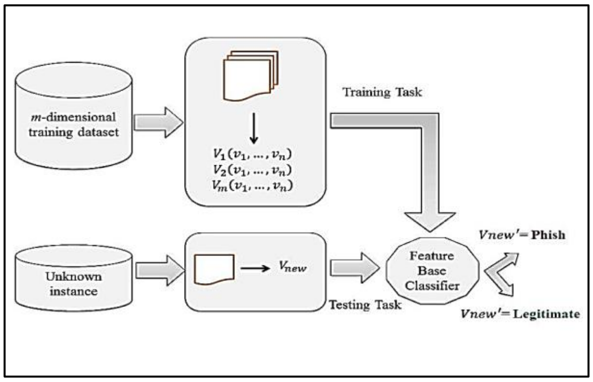


Figure 3 Typical Classification scenario with Training and Testing tasks as adopted in [35-37])

2.5 Evaluation Metrics

To demonstrate the outcomes of the proposed features and their categories to phish website detection performance, some primarily used metrics in the domain of phishing detection are utilized as they presented herewith. These metrics include *TP, FP, FN, Precision, Recall, F1-measure*. *True Positive* (*TP*) indicates the rate of correctly classified phish instances. The *False Positive* (*FP*) refers to the rate legitimate instances wrongly classified as the phishing ones. Whereas *False Negative* (*FN*) indicates the phish instances wrongly labeled as the legitimate ones. Each of the *Precision*, *Recall* and *F-measure* were computed through the parameters of *TP, FP* and *FN* as per Equations 5, 6 and 7, respectively. The maximal value of *Precision* shows the maximal positive webpages that were classified. Whilst the maximal *Recall* value denotes the minimal prediction error. Then, *F-measure* was used to compute the mean value of both measures and denotes the initial phishness indication of the extracted features [33-37] as follows:

$$Precision = \frac{|TP|}{|TP| + |FP|} \qquad (5)$$

$$Recall = \frac{|TP|}{|TP| + |FN|} \qquad (6)$$

$$F - measure = 2 \times \frac{Precisio \times Recall}{Precision + Recall} \qquad (7)$$

3.0 RESULTS AND DISCUSSION

The main aim of any hybrid anti-phishing technique is to explore more sophisticated features, propose predictive ones, and deploy them for distinguishing phish and legitimate websites. To this end, this section presents the strategy that this research pursued to extract, classify and assess the proposed features as illustrated in Figure 4. Two main steps were followed: features extraction step, and phishness induction step. In between, an assessment step was included to assess the extracted features with respect to their prediction susceptibility for phishing classification using co-occurrence criterion and machine learning classifier on the experimental dataset.

Features extraction step deals with analysing the website content to extract the wanted features from the relevant parts and components of any input website as aforesaid in Section 3.0 (Table 3). While, phishness induction step, applies the machine learning classifier (SVM) to learn the given feature matrix. This could approve the decision on whether the newly extracted hybrid features are able to distinguish the input websites as a novel or legitimate phish. Mainly, most of anti-phishing techniques as those were discussed here were trying to map an input web data to an output data using a specific induction rules. Experimentally, the most publically known machine learning and data mining tool WEKA from the Waikato Environment for Knowledge Analysis (WEKA) was utilized to apply SVM classifier.

3.1 Experimental Dataset

Generally, a preliminary set webpages including 16000 webpages of different exploits with 9600 living phish and 6400 legitimate webpages. Webpages were collected during 1st September 2015 and 1st November 2016 from different sources. All webpages were retrieved from three publically available data repositories; Phish Tank and Castle Cops (for valid phish and non-valid phish web pages), and Alexa's top sites archive (for legitimate web pages). Such data sources were commonly used in the literature due to some reasons. In PhishTank, the novel phish websites were reported periodically, but some of them were inaccessible once their short life span was expired.

Whereas, phish webpages reported by CastleCops archive included old and novel phishes whose source code files could be accessible and rendered [6, 24]. Also, some non-valid phish web pages were retrieved from datasets adopted in recently published works [16-24]. This is due to their possibility to encompass novel phish variants since the referring related works stated that they were classified as suspicious web pages (Yet, not validated as phishes).

Particularly, as enlisted in Table 4**,** the collected webpages involved imbalanced volumes of phish to non-phish webpages. Also, they varied in terms of their exploits like login forms, homepages, end-up web pages, redirecting web pages. The collected web pages involved those hosted in URLs of Chinese and French websites with Chinese and French languages as well as those in English language. Moreover, the collected web pages involve those of the most targeting industries by phishers such as financial organizations, retail services, payments services, governmental organizations and social networking.

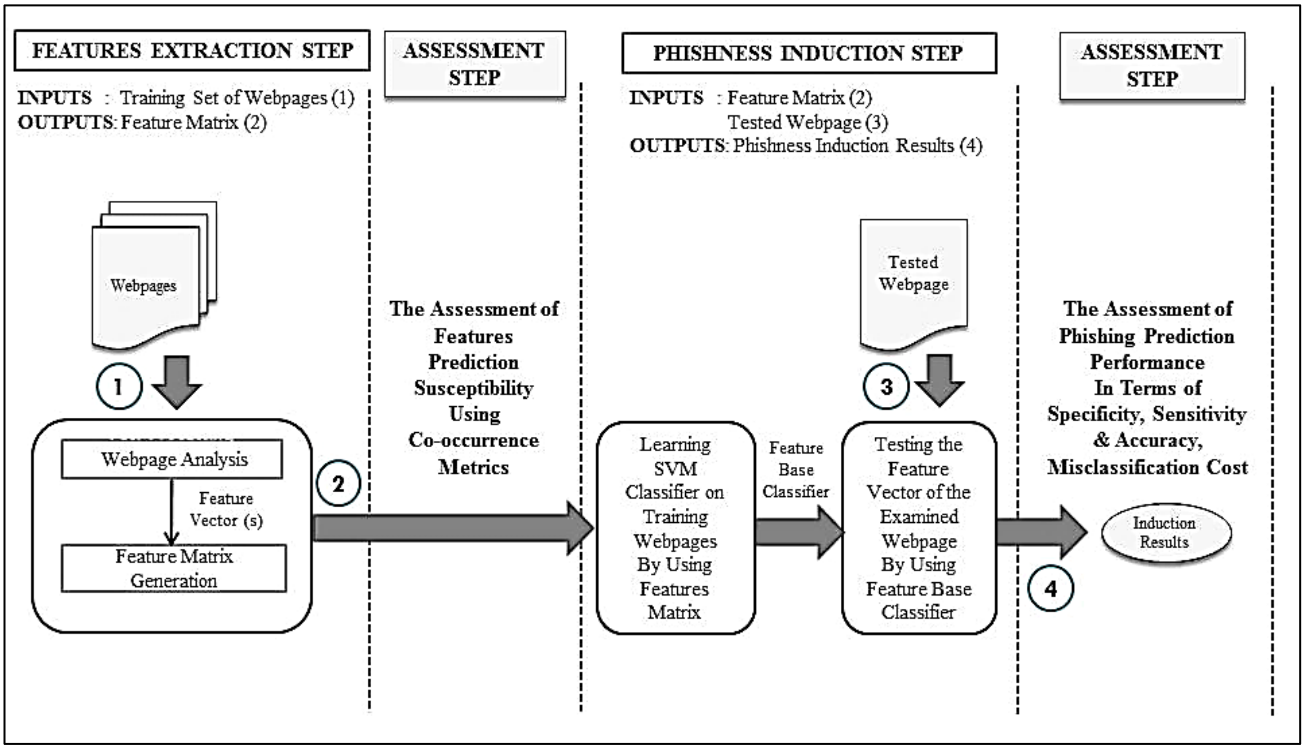

FEATURES EXTRACTION STEP
INPUTS : Training Set of Webpages (1)
OUTPUTS: Feature Matrix (2)
Webpages
1
Webpage Analysis
Feature Vector (s)
Feature Matrix Generation
2
ASSESSMENT STEP
The Assessment of Features Prediction Susceptibility Using Co-occurrence Metrics
PHISHNESS INDUCTION STEP
INPUTS : Feature Matrix (2)
Tested Webpage (3)
OUTPUTS: Phishness Induction Results (4)
Learning SVM Classifier on Training Webpages By Using Features Matrix
Feature Base Classifier
Tested Webpage
3
Testing the Feature Vector of the Examined Webpage By Using Feature Base Classifier
4
ASSESSMENT STEP
The Assessment of Phishing Prediction Performance In Terms of Specificity, Sensitivity & Accuracy, Misclassification Cost
Induction Results

Figure 4 Experimental strategy

3.2 Experimental Results

All the extracted hybrid features are enlisted in Appendix A in terms of their indexes, categories and values. The extracted features with their indexes were later used in the assessment of prediction susceptibility, and the novel phishness induction and its performance assessment. Extracted features are assessed in the form of their co-occurrence over the experimental collection of web pages (dataset). It is revealed that a feature of high co-occurrence score is expected to be crafted as novel phish feature.

Table 4 Description of experimental dataset

Web page Category	Web page Collection
Total number of instances	16000
Total number of non-phish instances	6400
Total number of phish instances	9600
Percentage of login form instances	33%
Percentage of redirecting instances	40%
Percentage of homepage instances	18%
Percentage of end up instances	9%
Percentage of English instances	90%
Percentage of French instances	2.67%
Percentage of Chinese instances	6.24%
Percentage of financial organization	34.4%
Percentage of payment services	32.1%
Percentage of retail services	15.1%
Percentage of social networking	11.2%
Percentage of governmental instances	4.6%

Co-occurrence scores of extracted features are enlisted in Table 5 along with their relevant features that were indexed (from F_1 to F_{58}) and categorized into three feature groups: Group1 for embedded objects category, Group2 for XSS category, and Group 3 for language independent features. Figure 5 plots the trendline of all features along with their relevant groups to showcase the diversity of their frequency over the experimental dataset.

On the other hand, the extracted features showed up divergent contributions in the form of their classification potentials such that each feature group showed particular classification rates on the experimental dataset. Moreover, all features of the examined groups are compacted to the group of hybrid features (Group 4) which was also learnt over the experimental dataset. This experiment was conducted to investigate the discriminating power of Group4 and whether it may surpass its competitors in phishing classification. Experimental results are plotted in graphs of Figures 6 and 7.

3.3 Discussion

Co-occurrence scores of examined features as presented in Table 5, and their variations as clearly shown in Figure 5, inferred the following observations:

i. F1 (Number of Scripting.FileSystemObject) in Group 1 (embedded objects-based features) implies counting the frequent use of an object, which executes the file system input and output on the user's computer. This object can be exploited by the phishers to control downloading and uploading the files to and from the computer during browsing and then distribute cookie files to the user's computer. Imitation and use of this feature has been highly occurred in both phish and suspicious websites so that it can be considered as the most significant feature for the novel phish detection. On the other hand, F24 is the feature with low occurrence rate in Group 1 (the number of out links), which implies it is highly occurring either in the phish websites or in the suspicious websites. However, its significance should be taken into account for further investigation to find whether or not it can be used to predict phishness through its combination with other features in one set.

Table 5 Counted occurrence of features due to their indexes and groups

Group1		Group2		Group3	
Index	**Value**	**Index**	**Value**	**Index**	**Value**
F1	0.8844	F1	0.63628	F1	0.962
F2	0.88295	F2	0.67426	F2	0.623
F3	0.87726	F3	0.75251	F3	0.453
F4	0.75251	F4	0.6228	F4	0.822
F5	0.73466	F5	0.7251	F5	0.562
F6	0.68119	F6	0.7234	F6	0.6211
F7	0.68107	F7	0.6418	F7	0.6331
F8	0.73436	F8	0.64778	F8	0.254
F9	0.6391	F9	0.63672	F9	0.7823
F10	0.64778	F10	0.6367	F10	0.6372
F11	0.64283	F11	0.64023		
F12	0.67426	F12	0.63951		
F13	0.66862	F13	0.63871		
F14	0.65541	F14	0.63744		
F15	0.64137	F15	0.63679		
F16	0.64023	F16	0.637093		
F17	0.71111	F17	0.63727		
F18	0.67369	F18	0.63675		
F19	0.65661	F19	0.63841		
F20	0.69789	F20	0.6382		
F21	0.6928	F21	0.6364		
F22	0.6605	F22	0.63639		
F23	0.64023	F23	0.63635		
F24	0.6341	F24	0.63732		

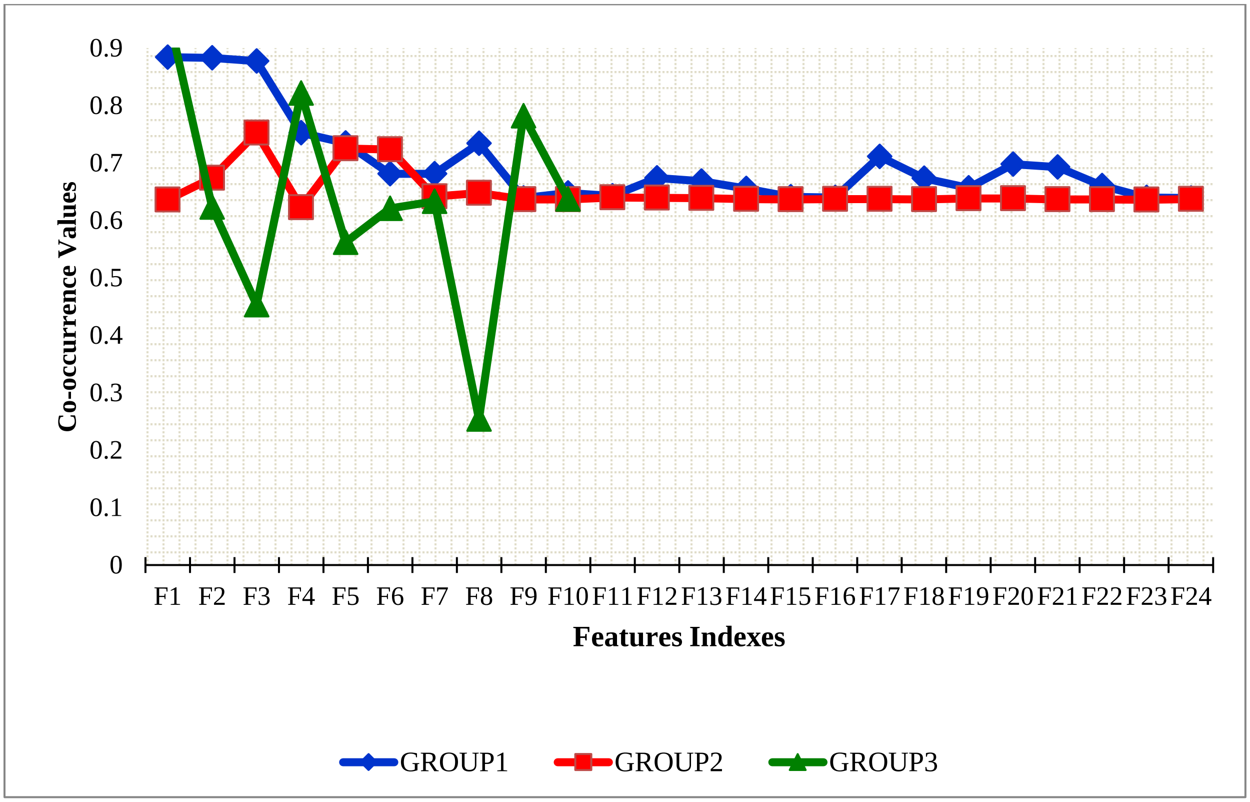
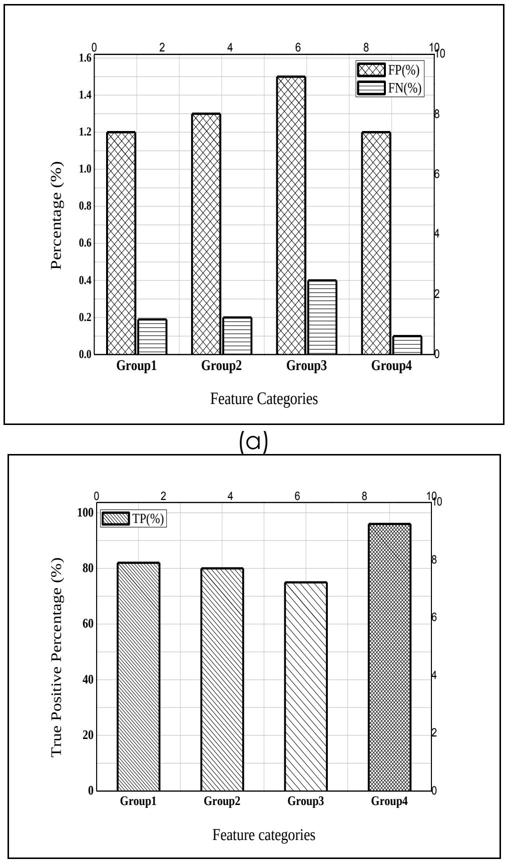


Figure 5 Prediction susceptibility of features with respect to their co-occurrence values over the experimental dataset



(a)

(b)

Figure 6: Phishness induction of feature categories: (a) FP and FN, and (b) TP; where Groups 1, 2, 3, and 4 refer to embedded objects features, XSS based features, language independent features and hybrid features, respectively

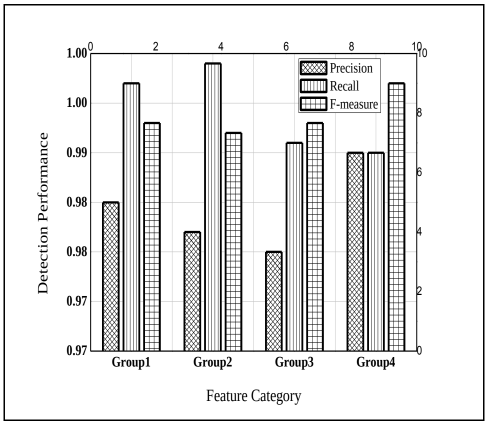


Figure 7 Detection performance of feature categories; where Groups 1, 2, 3, and 4 denote embedded objects features, XSS based features, language independent features and hybrid features, respectively

ii. Over the three investigated groups of features, Groups 1 includes features with higher occurrence rate than those in other groups. This implies that presence of some features in Group 1 is highly important for further investigation as a deceptive feature that might be exploited in novel phishes. This is due to the fact that most of the targets of phishes are website logins, which enable the phishers to acquire the user's credentials.

iii. These websites may deploy the embedded objects, scripts and components to develop replicas of the legitimate websites and redirect their visitors to the fake websites by including external links and modifying some attributes to the input passwords. In addition, they may be used to download suspicious files, codes and cookies from the Internet, executing ActiveX controls as a class ID of some built-in objects, executing shell instructions during web browsing, compromising webpages and redirecting users to the exploited servers.

iv. Some features included in Group 2 have high occurrence rate, which implies that phishes can exploit them to inject packed and obfuscated scripts using a client side scripting language, such as the JavaScript. For example, F3 in Group 2 can be exploited to inject loops that execute decode routines. Whereas the other features in the same group have somewhat less high values than those of the highest features. Thus, these features might be suggested as significant features to detect novel phishes.

v. Among all the extracted hybrid features, features F1, F2 and F3 in Group 1; F3, F5 and F6 in Group 2, and F1, F4 and F9 in Group 3 are the most occurring features. The second most occurring features are F8, F17, F20, F21, F2, F1, F7, F2, F6 and F7 from Groups 1, 2 and 3, respectively. But features F10, F15, F3 and F8 in Groups 1 and 2 are the least significant. This indicates that phishes and suspicious websites tend to contain the most occurring features, which are mostly non-traditional features. Thus, such features can be considered as novel features that are important for the novel phish prediction. These features need further investigation and evaluation to generate novel phish profile which could be done as the next research work.

vi. The rest of extracted features in Group 2 are mostly equal in occurrence and their occurrence probabilities tend to be more significant than those of the other groups, i.e. values of features occurrence in Groups 1 and 3 varied from 0.2 to 0.9. That implies the majority of the novel phishes may exploit these features for much sense, trickery and functionality.

vii. For the features in Group 3, it is obvious that phishes and suspicious websites tend to do some modifications in the URL's domain name with TLDs, dots and encoded domain names to imitate URLs of the legitimate websites. These features usually exist in most of the phishes and suspicious websites. Furthermore, Group 3 includes less number of features with different occurrence rates. Features such as F1 (Multiple TLD), F4 (Coded URL) and F9 (Number of dots in URL) have the highest rate of occurrence, because they can be used to refer to domains that serve the fake websites.

On the other hand, statistics and charts plotted in Figures 6 and 7, revealed the following observations:

i. Due to the low rate of TP and the high rates of FP and FN, features of Group 3 achieved the least F-measure value among all the examined groups of features. This implies that the rate of missing phish URLs increased up to 21.97%, because the phishers exploit legitimate URLs to upload their novel phish websites. Thus, sole use of the features in Group 3 is insufficient to leverage features of the novel phishes.

ii. Features of Group 1 were the most contributing features due to their F-measure value, which denotes their effectiveness in prediction. Group 2 composed of features with less contribution than those of Group 1, whereas the URL features of Group 3 had the least contributions.

iii. Features in Group 1 achieved higher prediction accuracy (i.e. F-measure) with relatively minimum rate of FP and FN. This infers that these feature types could be improved towards ranking their detection capability against the novel phishes to decrease both FP and FN. However, in practice, it is hard to perfectly predict novel phishness based on one type of feature, which may be the result of impure analysis of features exploited by all the novel phish websites.

iv. Combining all the extracted features in Group 4 reveals that they could increase the prediction accuracy without compromising the rates of FP and FN. The reasonable rate of FP and FN was achieved due to the use of multiple types of features that did not entirely overlap. Furthermore, the prediction accuracy could be improved by filtering all of these features into a subset of features that were fewer in number and the highest in relevance to the novel phishness indication.

v. Overall, results demonstrate the potential discrimination and holistic characterization that a large scale set of hybrid features could afford to outfit novel and prevalent phish website classification

In short, the aforesaid observations implied that despite of their frequently occurrence over the experimental dataset, examined features are varied in their co-occurrence values. Therefore, those investigated features should be taken into account in the classification of novel phish websites because they may be crafted in phish websites frequently for deception purpose in the future case of web data.

3.4 Comparative Analysis

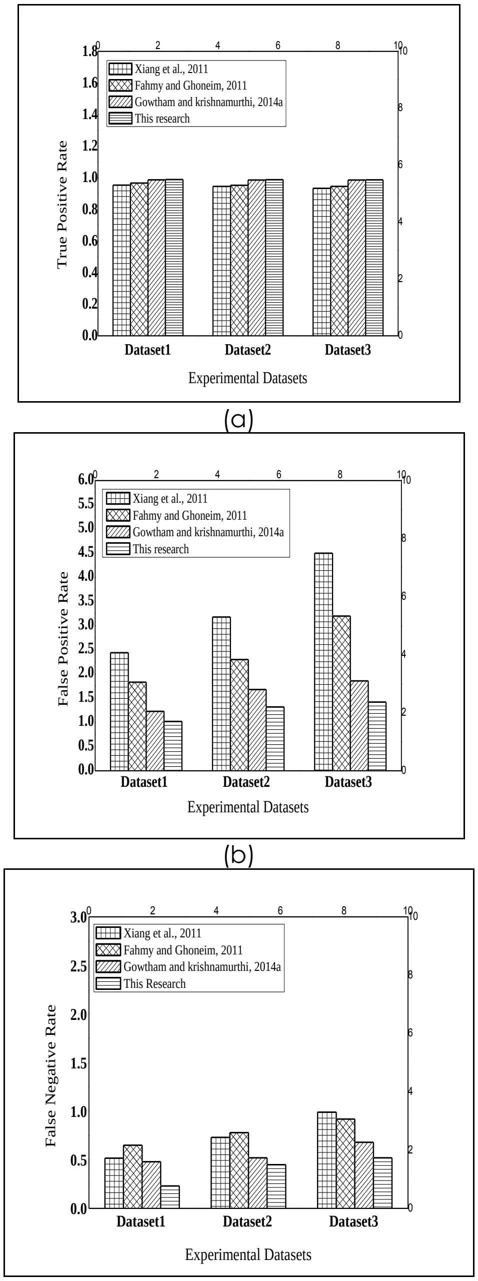
Some related works adopted different sets of generic phishing features to detect phish websites but their proposal did not contribute well to novel phishness prediction. To restate the difference in features and their classification outcomes between our work and some of the most renowned related works, Table 6 gives an overview of the previously used features and the proposed ones with respect to the number of features, and their categories. Regarding Table 6, it is observed that fewer and more different phishing features were investigated by the comparable works [11, 22, 23] in contrast to features proposed in this paper. In [11, 22, 23], the generic features were normally extracted from the URLs and HTML documents or the sources codes of the phish webpages. Moreover, 3rd party features are also used as supplementary features to determine discrepancies between the phish and legitimate websites with the aid of external resources and search engines for page ranking, verification of domain name systems (DNSs), and target website identification (WHOIS). Such features may not contribute the characterization of novel phish website due to their inconsistencies, computational costs, and their evasion by novel phishes.

Table 6 Comparison overview between this work and the related works in the form of deployed features

Feature type	[11]	[22]	[23]	This work
HTML	2	8	5	24
JavaScript	0	0	0	24
URL	10	9	7	10
3rd Party	3	1	3	0
Total	15	18	15	58

Regarding Table 6, the authors in [11] used two HTML features extracted from the input and login form as well as ten URL features with three features related to the third parties (e.g. WHOIS and Google page rank). In [22], nine URL features and eight HTML features were extracted from the title, image, input and text tags. They were used with one page rank feature by Google search engine. On the other hand, researchers in [23] proposed five HTML features, such as those related to the input and login forms, seven URL features related to patterns, symbols, number of dots, sensitive words, IP address and multiple TLDs as well as those extracted using the page rank of search engine as the third party features. Whereas, this work particularly identified 58 hybrid features extracted from the nodes of DOM tree and they include some native functions, attachment events, methods, attributes and other elements related to the embedded components and scripts in-lines of the website's source codes.

With the perspective of phishing classification outcomes, an assessment was attained across the experimental dataset by using our proposed features and those used in the aforesaid comparable works as shown in Figure 8. Figure 8 plots the classification outcomes in terms of the rates of false positive (FPR) and false negative (FNR) as well as true positive rate (TPR). Also, the classification outcomes are reported across the experimental dataset which was divided into three datasets: Dataset1 for training, Dataset2 for validating and Dataset3 for testing.



(a)

(b)

(c)

Figure 8 phishing classification outcomes of the proposed hybrid set of features versus the features sets adopted by the comparable works

Results plotted in Figure 8, showcase that the proposed features in this work achieved the best cases of TPR, FPR and FNR in contrast to those features adopted in [11, 22, 23]. This is due to their large number, their newly deployment, and their hybridity which attained more holistic prediction on almost phish websites on the experimental dataset. Indeed, more sophisticated features of hybrid feature category such as embedded objects features, XSS based features, and language independent features could successfully classify novel phish web websites. Despite the proposed hybrid set of features, the features set adopted in [11] reported lower TP with higher FP and FN because of its smaller number of features. Moreover, such features have less divergence with respect to their types and features particular categories that might fall short to detect novel phishes effectively on the dataset. Besides, the experimental dataset is of imbalanced class distribution in phishing and legitimate websites so that new phishing deceptions can be misclassified. Therefore, the features set of [11] lacked adaptation to novel phish websites. Whereas, the features set that adopted in [22], maintained a relatively noticeable rise due to their partial classification of phish websites across the datasets that varied in their class distribution and web page exploits. However, their results revealed their inability to characterize more advanced phishing deceptions. Similarly, features adopted by Xiang *et al.* [23] yielded the worst case of classification performance across the datasets amongst those of other related works. Such unacceptable classification outcomes among the others pointed out the partial characterization of the aforesaid features against novel phish websites as well as their partial covering up newly collected, larger in size, and more varied in class distribution datasets.

3.5 Future Work

Based on the aforesaid observations, this sub-section gives an insight on facets to be considered for future research:

i. Exploring and employing new phishing features can possibly yield valuable information to detect novel variants of phishes used by the phishers to bypass the existing anti-phishing campaign. Also, they will provide significant drive to the Internet phishing mitigation.

ii. Highly contributing features are extracted from the website content, such as the functional components, objects, elements, native functions of scripting language. These features are highly expected to be exploited by the phishers due to their functionalities in modification, imitation, redirection and injection of codes and links towards obtaining the user's confidential information. Furthermore, they could be assigned to more than one type of novel phishes.

iii. It is observed that the use of hybrid features in Group 4, could improve the detection accuracy against the novel phishes. Features of Group 4 perfectly indicate the effects of hybrid features on increasing the phishness prediction accuracy due to the reduced overlap and increased assignment of the hybrid features to more than one pattern of novel phish.

iv. The major issue that should be considered in using the hybrid features is the computational time and cost. Through the machine learning algorithms or classifiers, the use of 58 hybrid features may have negative impacts on the time and execution during extraction, training and testing over large data sets.

v. Another issue that should be considered is that phishness prediction using a classifier like SVM mainly depends on the web content quality and quantity of the collected data set. More precisely, the websites used for the training on the classifier could represent the deceptive features exploited in either collected phish or suspicious websites. Thus, the data set may affect the accuracy of classifier and thereby, the results of the novel phishness prediction.

vi. To avoid negative impacts caused by the dimension of features space (58 hybrid features), further improvements are needed to reduce the dimension of features space as well as the complexity and time of phishness prediction over a large data set in real life application.

vii. The above-mentioned empirical features assessment approach yields that all extracted hybrid features can be adopted for the novel phishness prediction, which is the major trend of the current research trend. But some of these features may have negative impacts on the overall detection, including the computational cost, specificity and sensitivity, due to their difference in significance. Thus, further evaluation is needed to identify the optimal combination of these features and obtain a set of potential features that are hard to exploit by the phishers for bypassing the existing anti-phishing campaign.

Overall, they principal inductive factor of a classification based detection method is the features in use. Moreover, to optimize the classification based detection method for novel phish website detection, both holistic characterization on phishing deceptions and accurate classification on novel phish patterns are the most striking merits that should be considered carefully. To attain these merits, novel, hybrid, and discriminative features are required to explore and assess frequently.

As such, the detection method could yield effective performance in the real time experience. To this end, we believe that our proposed features as demonstrated experimentally are promising to extend and optimize those classification based detection methods of limited detection scope. Furthermore, the introduced criteria could promote the assessment and the selection of the most discriminative features from a large scale space of hybrid features testified herewith experimentally. Consequently, a phishing filtering engine could be upgraded in terms of its inductive parameters and its performance toe configure resilient defense against evolving phishes.

4.0 CONCLUSIONS

One of the main motivations behind conducting this work, was to introduce new features (58 features) for more holistic characterization of phish website patterns and more accurate classification of the novel ones. Other motivation was to investigate the causality between the proposed features (as the primary inductive factors) and the performance of classification based detection method. All experimentations, assessments and their relevant findings gave a proof-check of features' contributions to phishing classification due to their high exploitations on phish websites and their prediction susceptibility on novel phish websites.

Accordingly, the proposed features are recommended as promising factors to extend the limited scope of the currently available phishing filters and improve their performance with minimal misclassification error on novel phish websites. On the other hand, pursuing the introduced assessment criteria could also promote the detection strategy with the perspective of features selection. They could produce the most discriminative compactness of features for more accurate classification results. Based on the presented findings, it was found that the hybridization of features of multiple categories would complement their contributions on effective phishing classification. Moreover, it was revealed that features varied in their discriminating power on phishing classification as well as their diversity in phishing frequent exploitations. Such observation gives a global view on two important facets: (i) hybrid features provide holistic description of phish websites that crafted in different types of features and then minimize the rate of false detections, (ii) selecting the most discriminative features is merely significant to obtain more accurate detection results and to minimize false detections, and (iii) exploring new features frequently is essential to obtain effective detection outcomes and thwarting novel phish websites.

Thus, further improvements can be made to select the most contributing features and discarding the least contributing ones with the aid of feature selection method for more efficient anti-phishing technique.

Acknowledgement

We are grateful for the UTM support to accomplish this dedicated effort.

References

[1] Khonji, M., Iraqi, Y. and Jones, A. 2013. Phishing Detection: A Literature Survey. *Communications Surveys and Tutorials, IEEE.* 15(4): 2091-2121.

[2] Purkait, S. 2012. Phishing Counter Measures and Their Effectiveness-Literature Review. *Information Management and Computer Security.* 20(5): 382-420.

[3] Gowtham, R., Krishnamurthi, I. and Kumar, K. 2014. *An Efficacious Method for Detecting Phishing Webpage Through Target Domain Identification. Decision Support Systems.*

[4] Rader, M. A. and Rahman, S. S. M. 2013. Exploring Historical and Emerging Phishing Techniques and Mitigating the Associated Security Risks. *International Journal of Network Security and Its Applications.* 5(4).

[5] San Martino, A. and Perramon, X. 2010. Phishing Secrets: History, Effects, Countermeasures. *IJ Network Security.* 11(3):163-171.

[6] He, M., Horng, S.-J., Fan, P., Khan, M. K., Run, R.-S., Lai, J.-L. and Sutanto, A. 2011. An Efficient Phishing Webpage Detector. *Expert Systems with Applications.* 38(10): 12018-12027.

[7] Whittaker, C. , Ryner, B. and Nazif, M. 2010. March. *Large-scale Automatic Classification of Phishing Pages. In NDSS.10*

[8] Kordestani, H. and Shajari, M. 2013. An Entice Resistant Automatic Phishing Detection. *5th Conference in Information and Knowledge Technology (IKT).* 134-139.

[9] Shahriar, H. and Zulkernine, M. 2012. Trustworthiness Testing of Phishing Websites: a Behavior Model-Based Approach. Future Generation Computer Systems. 28(8): 1258-1271.

[10] Islam, R. and Abawajy, J. 2013. A Multi-tier Phishing Detection and Filtering Approach. *Journal of Network and Computer Applications.* 36(1): 324-335.

[11] Gowtham, R. and Krishnamurthi, I. 2014. A Comprehensive and Efficacious Architecture for Detecting Phishing Webpages. *Computers and Security.* 40: 23-37.

[12] Barraclough, P., Hossain, M., Tahir, M., Sexton, G. and Aslam, N. 2013. Intelligent Phishing Detection and Protection Scheme for Online Transactions. *Expert Systems with Applications. 40(11):* 4697-4706.

[13] Aburrous, M., Hossain, M., Dahal, K. and Thabtah, F. 2010. Associative Classification Techniques for Predicting e-Banking Phishing Websites. *2010 International Conference on Multimedia Computing and Information Technology (MCIT).*

[14] Olivo, C.K., Santin, A.O. and Oliveira, L.S. 2013. *Obtaining the Threat Model for E-mail Phishing. Applied Soft Computing.* 13(12): 4841-4848.

[15] Alnajim, A. and Munro, M. 2009. An Anti-Phishing Approach that Uses Training Intervention for Phishing Websites Detection. *Proceedings of the 2009 Sixth International Conference on Information Technology: New Generations-Volume.*

[16] Pan, Y. and Ding, X. 2006. Anomaly Based Web Phishing Page Detection. *In Computer Security Applications Conference (ACSAC'06).* 381-392.

[17] Joshi, Y., Saklikar, S., Das, D. and Saha, S. 2008. PhishGuard: A Browser Plug-in for Protection from Phishing. *2nd International Conference in Internet Multimedia Services Architecture and Applications (IMSAA 2008).* 1-6.

[18] Likarish, P., Jung, E., Dunbar, D., Hansen, T. E. and Hourcade, J. P. 2008. B-apt: Bayesian Anti-Phishing Toolbar. *IEEE International Conference in Communications (ICC'08).* 1745-1749.

[19] Yue, C. and Wang, H. 2010. BogusBiter: A Transparent Protection Against Phishing Attacks. College of William and Mary. *ACM Transactions on Internet Technology.* 10(2).

[20] Shahriar, H. and Zulkernine, M. 2010. PhishTester: Automatic Testing of Phishing Attacks. *Fourth International Conference in Secure Software Integration and Reliability Improvement (SSIRI).*198-207.

[21] Prakash, P., Kumar, M., Kompella, R. R. and Gupta, M. 2010. *Phishnet: Predictive Blacklisting to Detect Phishing Attacks, In Proceedings of INFOCOM. IEEE.* 1-5.

[22] Fahmy, H. M. and Ghoneim, S. A. 2011. PhishBlock: A Hybrid Anti-Phishing Tool. *International Conference in Communications, Computing and Control Applications (CCCA).* 1-5.

[23] Xiang, G., Hong, J., Rose, C. P. and Cranor, L. 2011. Cantina+: A Feature-Rich Machine Learning Framework for Detecting Phishing Web Sites. *ACM Transactions on Information and System Security (TISSEC).* 14(2).

[24] Han, W., Cao, Y., Bertino, E. and Yong, J. 2012. Using Automated Individual White-List to Protect Web Digital Identities. *Expert Systems with Applications.* 39(15): 11861-11869.

[25] Gastellier-Prevost, S., Granadillo, G. G. and Laurent, M. 2011. Decisive Heuristics to Differentiate Legitimate from Phishing Sites. *Conference in Network and Information Systems Security (SAR-SSI).* 1-9.

[26] Zhuang, W. , Jiang, Q. and Xiong, T. 2012. An Intelligent Anti-Phishing Strategy Model for Phishing Website Detection. *32nd International Conference in Distributed Computing Systems Workshops (ICDCSW).* 51-56.

[27] Alkhozae, M. G. and Maratfi, O. A. 2011. Phishing Websites Detection Based On Phishing Characteristics in the Webpage Source Code. *International Journal of Information and Communication Technology Research.*

[28] Nguyen, L. A. T., To, B. L., Nguyen, H. K. and Nguyen, M. H. 2013. Detecting Phishing Web Sites: A Heuristic URL-Based Approach. *International Conference in Advanced Technologies for Communications (ATC).* 597-602.

[29] Zhang, J. and Wang, Y. 2012. A Real-Time Automatic Detection of Phishing URLs. *2nd International Conference in Computer Science and Network Technology (ICCSNT).* 1212-1216.

[30] Basnet, R. B. and Sung, A. H. 2012. *Mining Web to Detect Phishing URLs. 11th International Conference in Machine Learning and Applications (ICMLA).* 568-573.

[31] Cardoso, E. , Jabour, I. , Laber, E. , Rodrigues, R. and Cardoso, P. 2011. An Efficient Language-Independent Method to Extract Content from News Webpages. *In Proceedings of the 11th ACM symposium on Document engineering.* 121-128.

[32] Khonji, M., Iraqi, Y. and Jones, A. 2011. Lexical URL Analysis for Discriminating Phishing and Legitimate Websites. *In Proceedings of the 8th Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference. ACM.*

[33] Uzun, E. , Agun, H. V. and Yerlikaya, T. 2013. A Hybrid Approach for Extracting Informative Content from Web Pages. *Information Processing and Management.* 49: 928-944.

[34] Fu, L., Meng, Y., Xia, Y. and Yu, H. 2010. Web Content Extraction Based On Webpage Layout Analysis. *2nd International Conference in Information Technology and Computer Science (ITCS).* 40-43.

[35] Wang, H. , Zhu, B. and Wang, C. 2012. A Method of Detecting Phishing Web Pages Based On Feature Vectors Matching. *Journal of Information and Computational Systems.* 9: 4229-4235.

[36] Lakshmi, V. S. and Vijaya, M. 2012. Efficient Prediction of Phishing Websites using Supervised Learning Algorithms. *Procedia Engineering.* 30: 798-805.

[37] Huang, H. , Qian, L. and Wang, Y. 2012. *A SVM-based Technique to Detect Phishing URLs. Information Technology Journal.* 11: 921-925.

Appendix A

Appendix A Extracted hybrid features with their relevant indexes and groups

Group (1): embedded objects features		**Group (2): XSS features**	
Index	**Feature**	**Index**	**Feature**
F1	Number of Scripting.FileSystemObject	F24	Number <input> in java scripts
F2	Number of Excel.Application	F25	JavaScript scripts length
F3	Presence of WScript.shell	F26	Number of functions' calls in java scripts
F4	Presence of Adodb.Stream	F27	Number of script lines in java scripts
F5	Presence of Microsoft.XMLDOM	F28	Script line length in java scripts
F6	Number of <embed>	F29	Existence of long variables in JS
F7	Number of <applet>	F30	Existence of long function in JS
F8	Number of Word.Application	F31	Number of fromCharCode()
F9	link length in <embed>	F32	Number attachEvent()
F10	Number of <iframe>	F33	Number of eval()
F11	Number of <frame>	F34	Number of escap()
F12	Out-of-place tags	F35	Number of dispacthEvent()
F13	Number of <form>	F36	Number of SetTimeout()
F14	Number <input>	F37	Number of exec()
F15	Number of MSXML2.XMLHTTP	F38	Number of pop()
F16	Frequent <head>, <title>, <body>	F39	Number of replaceNode()
F17	<meta index.php?Sp1=>	F40	Number of onerror()
F18	"Codebase" attribute in <object>	F41	Number of onload()
F19	"Codebase" attribute in <applet>	F42	Number of onunload()
F20	"href" attribute of <link>	F43	Number of <script>
F21	Number of void links in <form>	F44	frequent<div onClick=window.open()">
F22	Number of out links	F47	Number of onerror()in javascripts
F23	Number of <form> in java scripts	F48	Number of SetInterval()
Group (3): Language independent features			
Index	**Feature**		
F49	Multiple TLD		
F50	Brandname in hostname		
F51	Special symbols in URL		
F52	Coded URL		
F53	IP address instead of domain name		
F54	Typos in Base name		
F55	Long domain name		
F56	Misleading subdomain		
F57	Number of dots in URL		
F58	Path domain length		