Computer Science and Engineering 2026, 16(1): 1-12
DOI: 10.5923/j.computer.20261601.01

Enhancing Phishing Attacks Detection by Optimizing URL Features Selection in Machine Learning-based Approach

Sami El Flesh[1,*], Muhammad Abdullah[1], Abdulhamid Zaidi[2]

[1]Department of Computer Science, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Serdang, Malaysia
[2]Department of Electronics and Computer Engineering Technology, Bailey College of Engineering and Technology, Indiana State University, Terre Haute, Indiana, USA

Abstract Despite ongoing efforts to enhance phishing detection methods and strategies, the use of cut-off ratings has primarily served to improve classification accuracy. However, current research is shifting toward URL feature discovery and machine learning approaches. Most contemporary anti-phishing studies focus on proposing novel feature selection techniques or optimizing classification algorithms. To the best of our knowledge, there remains a lack of reliable, effective, and systematic frameworks for feature selection that address the persistent inaccuracies in phishing detection. This gap stems from the limitations of existing feature selection algorithms, which often lack a rank identifier to determine the most optimal and impactful set of baseline URL features for phishing classification. Furthermore, the absence of a structured approach to refining these baseline features for machine learning classifiers exacerbates the problem. In this paper, we propose an algorithm that introduces a feature cut-off rank identifier. This mechanism benchmarks and selects features that exceed a defined threshold, ensuring only the most relevant attributes are used for phishing classification. Additionally, we present a systematic feature selection framework that employs data perturbation and data function perturbation ensembles to generate optimized baseline features. Experimental results demonstrate that these optimized features, when integrated with a Random Forest classifier, can accurately distinguish between phishing and legitimate websites with an accuracy of approximately 97%.

Keywords Features selection, Machine Learning, Naïve Bayes, J48, Random Forest, Random Tree, REPTree, Spam email detection, SVM

1. Introduction

Technology evolution is rising quite rapidly with the tremendous expansion of digital age that provides us with all the privilege of being at the forefront of the ongoing endeavor that has the high potential to uplift and shape our lives and future generations around the globe, digitality connected technology enabled us to be more efficient by leveraging on online services; especially in the past few decades, where many services become available online for our convenient such as online shopping, online banking, social media, government services, and many more.

This evolution was the drive to the significant increase in online consumers and subsequently the phishing arracks are continuing to grow, with many instances still going unreported or underreported. In Figure 1 an illustration of a phishing campaign conducted in one of the oil services companies' operating in Malaysia on quarter basis rollout to evaluate the reporting and susceptibility for approximately 2800 employees. The obtained responses are categorized in 4 indictors as below:

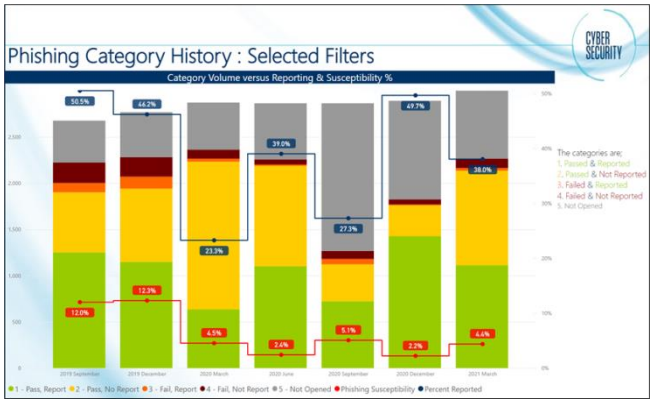

Figure 1. Phishing Campaigns Reporting and Susceptibility

* Corresponding author:
selflesh.edu@gmail.com (Sami El Flesh)
Received: Dec. 20, 2025; Accepted: Jan. 15, 2026; Published: Jan. 23, 2026
Published online at http://journal.sapub.org/computer

1) Green represents employees who successfully recognized the phishing email and proactively reported their suspicious.
2) Yellow represents employees that successfully recognized, but not reported it as potential phishing.
3) Orange represents employees who failed to recognize the phishing, but they have reported it.
4) Red represents employees who failed to recognize the phishing, but they did not report it.

The takeaway from the Figure 1 above those phishing attacks are still effective technique where all it takes is a failure of one user to recognize the phishing attack, Hence the necessity of shifting the overburden of detecting phishing attacks from the users to technology become a vital need.

We are also seeing that spear phishing and social engineering tactics are getting more sophisticated, more targeted, and more advanced. In this project our goal is to propose an automated mechanism for feature selection leveraging on machine learning techniques to further optimize an URLs features. Features' cut-off ranking is one of the algorithms that possibly will be utilized to automatically determine the optimal subset features and reduce irrelevant URLs features without losing the precision of the detection accuracy. With the proposed feature selection framework, we expected an increase of phishing detection accuracy, hence the weakest layer of defense "people" will be hardened leveraging on the machine learning automation to detect phishing attempts. Countless researchers have tried to different approaches and technologies to increase the phishing detection accuracy and prevent the phishing attacks from taking in place before it reaches end users. However, most of the developed and proposed systems exhibits some limitations and the lack of providing a complete systematic framework that automate the optimal URL baselines features identifier that can be the most effective dataset for machine learning.

Previous anti-phishing detection studies focus on proposing unprecedented features selection or optimized algorithms of phishing classification are insufficient to improve the classification accuracy. Developing an effective and a systematic technique for feature selection as well as an end-to-end framework to overcome the inaccuracy detection of phishing attacks was not on the top priority. As consequences, the inaccuracy problem remains one of the critical issues in detecting of phishing attacks in machine learning-based approach. Below are the two problems that contribute to this problem.

1) Incomplete feature selection method due to the lack of feature cut-off rank identifier to determine the most effective and tuned subset of baseline features for phishing attacks classification.
2) Ineffective feature selection framework due to the lack of systematic approach to optimize the baseline features that are most effective for a machine learning classifier.

Our main objective in conducting this project is to enhance the detection accuracy of phishing attacks by increase the number of instances that correctly classified as phishing.

1) To propose an algorithm that can identify the feature cut-off rank by benchmark and select only the features deemed to be higher than the cut-off rank threshold, then the identified features will be used for phishing attacks classification.
2) To propose an effective (systematic) feature selection framework by using data perturbation and data function perturbation ensembles in order to produce the optimized baseline features for a machine learning phishing attacks classifier.

2. Literature Review

In this section we will share some of the comprehensive reviews and detailed discussions that researchers already published in the form of journals or conference proceeding to shed more light on the problem we are addressing which is phishing detection accuracy. By reviewing related work in the phishing detection and the classification methods and techniques has been employed by researchers to distinguish the legitimate URLs from the phishing ones.

The risk of phishing exposure well known and recognized. In the 90s where a group of hackers called Warez community also known as first "phishers" have impersonated as American Online (AOL) employees and deceived the AOL's users to collect their personal information and login credentials. As a global leader in internet service provider at that time with approximately a million subscriber and online consumers of their services. This indeed led to drag the hacker's attention to conduct the phishing first seeds and since then the phishing tactics evolving until today. Thus, in order to better understand the evolution to phishing and anti-phishing techniques, we conducted an extensive literature review of the most relevant researches utilized URL features algorithms and machine learning.

Historically, there are many phishing detection techniques and approaches has been employed by many researchers over the time and technology evolvement since the evolution of phishing where by some of the proven approaches might not be relevant today and to cope with the fast evolving technologies and significant increase of internet and communications consumers, exploring more efficient and automated approached become essential to leverage on more relevant technologies such as machine learning. To be more aligned with the research objectives and the attended approach for our methodology, we can categorize the methods that been employed in detecting whether the URL is phishing or legitimate into none-machine learning techniques and machine learning techniques.

A. CONTENT AND SEARCH-BASED APPROACH

It is worth to mention that the main goal of phishing ULRs and webpages is to deceive online consumers and users around the global into making the websites they are consuming, and surfing look like a legitimate. The content-based analysis approach utilizing deep analysis of webpages

contents and extract features via classifiers algorithms and use other services such as DNS servers, and search engines. This method relies highly on the visual content similarities to distinguish between phishing and legitimate websites.

[1] have proposed a weighted classifier to decide if the words that extracted from HTML content and URLs, where those words might indicate any characterization of phishing setup for instance a brand name to replicate a legitimate website content. The weights are measured according to their location within the URLs. Then, the measured weights decoupled with the relevant term frequency-inverse document frequency (TF-IDF) weight, in a form a numeric presentation that illustrate the significant of any word to a document frequency, after that based on the world's highest probability a list of words sent to search engine to get the highest frequency domain name among the top thirty outputs. At the end, a decision will be made if legitimate or phishing website by comparing the owners of nominated domain name which returned by WHOIS service. Although the proposed methodology seemed promising and giving the expected results, yet it's still limited to words contained in the websites and not utilizing the features lookup from the URLs.

In detecting phishing based on websites contents [2] proposed a methodology that analyze the logo images to identify whether if the relevant website identity is matching a phishing or legitimate website. The proposed solution consisted of logo extraction and identity verification. For the first step which is logo extraction as it is very difficult to distinguish which image is a logo and to have more realistic approach, they decided to download all the images by query the webpage and applied a subprocess to exclude any images unlikely to be a logo and for that the criteria defined as any image with one color scheme or any image with a width or height of less than 10 pixels. The in the second step which is verification of identity by utilizing Google Image search database the portrayed identify then retrieved. This allowed them to perform the identify comparison and validation and because domain name has relation with the logo, the domain name was considered the identity. Therefore, comparing the domain name returned from the query website with the one returned by Google Image search. The accuracy achieved by the proposed approach concluded as 93.4%. Yet, while the outcome looks superb, the True Negative Rate still low and giving the fact that it needs to download all the images to perform logo identification seemed a bit excessive computational process where other technologies might be more efficient for image process such as image recognition utilizes.

Focusing on more lightweight detection approach using search engines for phishing website [3] introduced the lightest possible attributes/features namely domain name and page title instead of loading the complete webpage only these two features can be extracted. An n intelligent anti-phishing chrome extension was developed named light weight phish detector (LPD). The developed extension also suggests the webpage authenticity on top of detection. The LPD used very light features to identify phishing websites, but it may result a false positive over evolved and recent launched websites where phishers indeed will find it easy to deceive such lightweight utilizing only two features.

B. FEATURE SELECTION AND MACHINE LEARNING APPROACH

The URLs features basically you could describe it as a combined number of attributes that form a unique path on the internet to reach out to a specific destination. This can be number of dots in the webpage URL, level of domain, URLs' length, HTTPs exist or not, number of '@', if '//' exists in the path of webpage URL. The feature selection and machine learning approach now days significantly utilized by many researchers in the recent researches due to the promising results of increased accuracy as well as the automation aspect that brought by the machine learning concept.

[4] conducted an evaluation comparison of few conventional feature selection methods to benchmark the performance of the feature selection methodology, which include correlation Based Feature Selection (CFS), filter measures such as (Relief-F and IG) and the wrapper method. According to results in findings the wrapper approach together with the best first forward search demonstrated an outperforms feature subset derived by Relief-F and Information Gain, whereas Correlation Based Feature Selection (CFS) showed the lowest performer. Also, the obtained results clearly indicate the effectiveness of predication and the consistent performance of Random Forest classifier compared to SVM and C4.5 classifiers. However, in some cases where performance come with a cost, in this case the computational overhead was a drawback for the wrapper method, which made it not the preferred method, hence the focus should be on optimizing the features filter measures to overcome the computational overhead concern and find less intensive computational methods.

Evaluating two common features selection methods is the research base conducted by [5] namely wrapper and Correlation Based Feature Selection (CFS), where an experiment has been performed to test feature space searching methods and to be more specific both generic algorithm and greedy forward selection were the candidates applied on features extracted from the webpage itself combined with search capabilities. In this experiment the Classifiers Logistic Regression, Random Forest and Naive Bayes were used to evaluate the feature subsets performance. The obtained results revealed that wrapper method achieved highest detection accuracy compared to CFS. Yet, the wrapper method as well known it is more computationally intensive, hence it was not the preferred approach in the applications of feature selection.

[6] have decided to evaluate 47 URL features to detect phishing URLs by using Chi-Square ranked values, Information Gain (IG) and correlation Based Feature Selection (CFS). The authors have observed a considerable reduction in the values of filter measures in the range of 20th to 21st features and have utilized this range gap as cut-off ranker in the values of filter measure to nominate top 20 features as baseline subset features. Although the results showed a stable detection accuracy when the nominated features subset are used, yet, while the results were showing a consistent and stable

performance, it is not very clear on how they achieved the identification of the cut-off ranker computational wise. Further experiments have been conducted by using only 12 ULR features that produced as an outcome of intersecting the features subsets of IG, Chi-Square and CF. As per the results an impact of only (0.28%) in average decrease in accuracy in comparison to the accuracy of the full feature set. Thus, the proposed intersecting features subsets approach demonstrated the effectiveness of optimizing the feature dimensionality without necessarily impacting the accuracy performance.

Similarly, by investigating the benchmark subset features [7] employed IG and Chi-Square to identify the benchmark of features subset for phishing detection. A systematic way to identify the cut-off ranks for features ranked by Chi-Square and IG was suggested by the authors. The proposed method was about threshold-based rule, where the cut-off ranks defined as a minimum two successive features with 50% variation in values of Chi-Square and Information Gain attributes rankers. The idea is to look at the value when the cut-off rank is triggered and compare the value against the recommended minimum value of filter measure. If it was below should be excluded.

Leveraging on Information Gain (IG) and C5.0 classifier, [8] have decided to use the ranking among 40 gathered from previous techniques to detect phishing and spam. The analysis done by the authors was mainly by utilizing Information Gain (IG) ranker to three different datasets to cross-rank the features, with an objective to identify the most effective subset of features. By applying the intersection function over the top 10 features that already ranked by IG. On the other hand, using the obtained three feature sets evaluated the accuracy utilizing C5.0 classifier and the results revealed that the feature sets with higher IG valued outperformed the feature sets with lower IG values. However, the assessed performance was very limited to only one filter measure, namely IG. Ideally to have more productive study a minimum and common filter measures should have employed to provide some good insights.

Using natural language processing together with machine learning a detection algorithm was proposed by [9] to perform a semantic based analysis of the content of the text to validate each sentence appropriateness. By applying Natural language processing (NLP) to extract and parse each sentence and based on the words analysis the algorithm tries to predict if the sentence is a command or a question, then the potential topics of the commands and questions are extracted by finding verb-direct object pairs and then each pair checked against topic blacklist database. The machine learning role is to generate the blacklist of suspected malicious pairs based on the training dataset. Yet, not enough experiments have been conducted and results only compared with Netcraft, also this approach relies on interception and analysis of texts in the email body which might not be a valid criteria after a while as the attackers become smarter every day and they learn from all the existing anti-phishing approaches.

[10] proposed a phishing detection approach by using a feature selection framework that automatically nominating the most optimized URL features among the extracted 48 features extracted from the URLs by an invented algorithm called CDF-g in which works to determine the top effective features in machine learning via patterns recognition of filter measures values, thus theoretically should not be explicitly work for a specific dataset and forming more adaptive approach to other datasets as well. [11] found that nominated features baseline resulted from the proposed approach yield to a promising outcome when the baseline dataset leverage on the Random Forest Classifier compared to other classifier used in the experiment by achieving (94.6%) of detection accuracy using (20.8%) of the original features, in addition of accuracy the results revealed that the proposed solution by [12] computationally more lighter given that only (20.8%) of the original feature scope utilized and yet the results still competitive compared to other features sections proposed by other researchers in the related work.

AdaBoost and MultiBoosting approach was proposed by [13] where every instance from the training dataset assigned with the associated equivalent importance by weighting the instance according to classifier's output aiming to decrease the correctly classified instances, while increasing the incorrectly classified instances. This will lead to generate 2 sets low importance and more challenging. Then a classifier will eb built for the categorized data sets with more focus on the more challenging instances. And the weights of each instance are augmented in accordance with the new classification performance, with this exercise increase the possibility to identify and reclassify more challenging instances to low importance category. Adaboost with SVM classifier outperformed in term of detection accuracy. Nevertheless, the dependency on webpages content requires more testing to prove the performance consistency.

Web pages contents extraction, URL feature extraction and high dimensionality optimization were the proposed technique by [14]. The main idea was to fine tune the extracted web pages contents and URLs features as a combination, then applying the high dimensionality minimization to reduce the variables set. Furthermore, the fins tuned dataset tested applying the machine learning classifier to measure the performance of the proposed model and compare it with results highlighted in related work section. The (SVM) support victor machine algorithm was the only classifier applied and the results reveals that the proposed model achieve better performance accuracy comparted to other classifiers. However, the results require further prove and investigation since the comparison used did not represents the actual performance since the results compared against neither used the same dataset nor the same classifiers.

While previous related work has focused on different techniques and approaches to tickle phishing detection accuracy, yet the efforts towards optimized feature selection frameworks still evolving. [14] and [15] leveraged on webpage contents technique. However, this approach is either comes with high dependency on webpage content or very limited features were included in the scope. Also, the webpage content approach incudes but not limited to visual analysis

which might be an overhead computational wise. Overall, the webpage contents technique might work very well, but from the effectiveness, automation, and computational standpoint this can be a limitation to adopt such techniques.

On the other hand [16] used natural language processing techniques and machine learning to analyze the email content and identify some keywords that potentially can help to classify the email if legitimate or phishing by checking it against a list of maintained sensitive words or sentences. However, the attackers keep evolving and learn from the existing anti-phishing techniques to evade such approach. Besides that, maintaining the list of words can be a hassle and of course another dependency that we should try to avoid.

[17] employed the attribute ranker approach to identify the baseline features. Only Information Gain (IG) was performed, and consistency nine features has been nominated across three different datasets, hence the obtained results not sufficient to conclude the achievement and effectiveness of the proposed framework.

3. Research Method

The main purpose of this chapter is to shed some light on the proposed research methodology and how it was conducted to accomplish to achieve the research objectives of this research. In this chapter the methodology will be presented in a flowchart form to illustrate the high level of overview, followed by elaboration about each phase have been utilized.

After extracting the baseline features dataset from the full features' dataset, then apply different features ranking algorithms on different classifiers to realize the effect of feature selection and evaluate its impact on different type of classifiers using default settings and parameters offered by Weka Tool without expensively drilling down to fine-tune the parameters of classifiers. Our proposed methodology as illustrated in Figure 2.

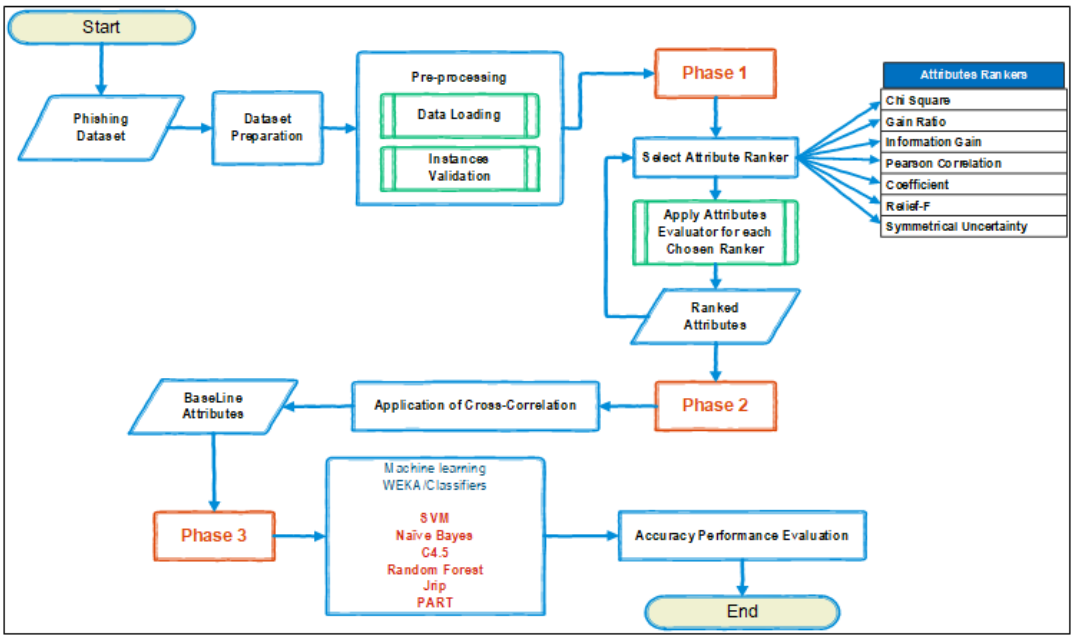

Figure 2. Methodology Flow-chart

A. PREPARATION OF DATASET

The same dataset complied by [18] will be utilized in the proposed workflow to have the same benchmark indicated in the proposed workflow. The dataset consists of 10,000 phishing instances samples equally distributed where 5,000 samples are legitimate URLs while another 5,000 samples are phishing as shown in Figure 3 below.

The dataset contains 48 attributes that are a representation of ULRs contents including but not limited to attributes such as how many dots in URL, counts of total number of characters in the URL, check if the "@" or "~" symbols are exist in the URL, counts the numbers of any of the special characters "&#%_" in the URL and etc.

The dataset already in ". ariff" format which is compatible with Weka tool and ready to be consumed in the planned experiment as explained in the methodology.

B. DATASET PRE-PROCESSING

Upload the full dataset with all the 48 ULR attributes using the Weka tool, then we generate a new dataset by including only the most effective features that identified by the Cumulative Distribution Function gradient (CDF-g) algorithm [19]. The new generated baseline-features dataset contains only (10 Attributes) as listed in Table 1 out of (48 attributes in total) in which will be utilized in our experiment along with the full dataset to gain some deep comparison of the nominated classifiers performance. The generated baseline features then will be used as our criteria and benchmark to measure the performance of our proposed methodology verses the HEFS (Hybrid Ensemble Feature Selection) reimplemented work and results by comparing the identified classifiers accuracy performance.

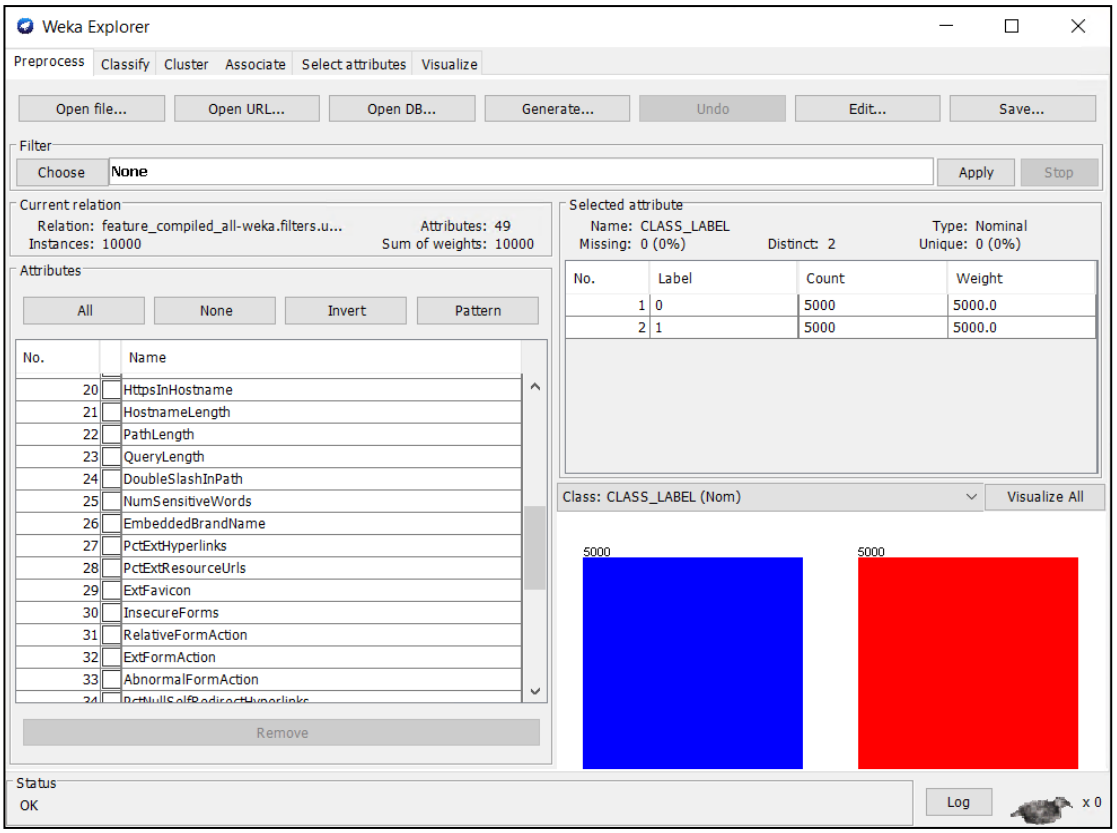

Figure 3. Dataset Loading and Validation

Table 1. HEFS Baseline Attributes

NO	Attributes	Description
1	FrequentDomainNameMismatch	Checks if the most frequent domain name in HTML source code does not match
2	PctNullSelfRedirectHyperlinks	Counts the percentage of hyperlinks fields containing empty value, self-redirect value
3	NumNumericChars	Counts the number of numeric characters in the webpage URL
4	PctExtHyperlinks	Counts the percentage of external hyperlinks in webpage HTML source code
5	NumDash	Counts the number of "-" in webpage URL
6	PctExtNullSelfRedirectHyperlinksRT	Counts the percentage of hyperlinks in HTML source code that uses different domain
7	PctExtResourceUrlsRT	Counts the percentage of external resource URLs in webpage HTML source code.
8	ExtMetaScriptLinkRT	Counts percentage of meta, script and link tags containing external URL in the
9	SubmitInfoToEmail	Check if HTML source code contains the HTML “mailto” function
10	NumSensitiveWords	Counts the number of sensitive words (i.e. "secure", "account", "banking", "webscr") in webpage URL

C. PHASE 1: ATTRIBUTES/FEATURES RANKERS

In this phase, the attributes evaluators rankers will be applied on the full dataset features to produce the ranked attributes list as per the evaluator algorithm. Ideally the outcome from each evaluator a list of the attributes sorted in order base on their weight and significance. We have employed six different attributes evaluators. Namely Chi-Square, Gain Ratio, Information Gain Pearson Correlation Coefficient, Relief-F and Symmetrical Uncertainty. Where each evaluator applied on the same dataset using Weka tool, by this we manage to generate the required list of attributes associated with each ranker.

As illustrated in the Figure 4 where the results of the Chi-square ranker showing the list of attributes in order based on the significance, for instance started with attribute #27 as most significant attribute followed by the retributes #28, #48, #34 and so on. The same step will be applied on the six rankers mentioned earlier to produce six different lists where the attributes orders will be different corresponding to selected ranker.

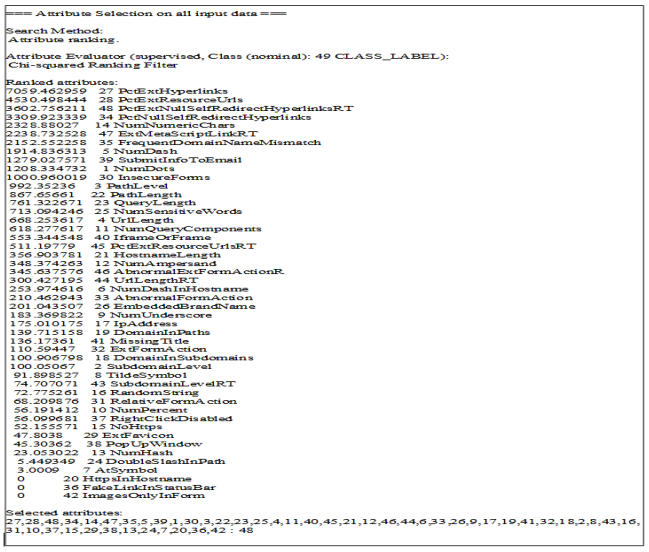

Figure 4. Example of Ranked Attributes Using Chi-square

D. PHASE 2: CROSS-CORRELATION

The proposed cross-correlation model considered as the main contribution and the most important phase in the proposed methodology whereby the cross-correlation model will be applied on the produced attributes lists from phase 1. The concept of the proposed cross-correlation model aiming to reproduce the most optimal baseline features that contributes significantly to phishing detection accuracy as well as significantly reducing the computational power when the nominated baseline features applied in real-world. In other words when using for instance 10 features instead of the whole 48 features while achieving the target accuracy will benchmark the success criteria to achieve the objectives in term of sustainable accuracy performance and reduced computational power. In order to optimize the proposed model performance, we have limited the scope to first 15 attributes from each ranked list considering that the first ranked 15 attributes will the most significant attributes contributing to phishing detection accuracy. The idea behind it is to compare each attribute from the first 15 attributes in each ranked list with the other five ranked lists and count the number of matches for each attribute among the other ranked lists where at the end of this phase a pre-final baseline list of 15 features along with the count of matches found in the other ranked lists will be produced. Then, by nominating 10 features based on the highest matches count giving that the highest matches count represents the significance of a particular attribute since most of the evaluator rankers ranked that attribute in the first 15 features. Figure 5 shows the cross correlation algorithm implemented in the proposed model.

An overview of the CCAR framework shown in Figure 6 where the six lists of the first 15 features obtained from the six attributes rankers will be the input. For each feature {F #1, F #2, …., F #15} in each list will cross correlated against feature match in the other 5 lists, if the feature match is TRUE add one to F_match_count and add the feature location to F_sum_order. The same process applied to all the 15 features in all the six lists to obtain a new list of nominated features along with their F_match_count and F_sum_order.

Input: L (List of Ranked Attributes), N (15 Attributes * 6 lists)

Output: BLF (Baseline Features List)

Begin:
For all I ∈ N do
  F_match_Count = 0
  F_sum_order = 0
  For all J ∈ N do
    If L(I) = L(J) then
      F_match_Count = F_match_Count +1
      F_sum_order = F_sum_order + feature_location_in_list
    End if
  End for
  If L(I) ∉ BLF
    Append L(I), F_match_Count, F_sum_order to BLF
  End If
End for
End

Figure 5. CCAR Algorithm

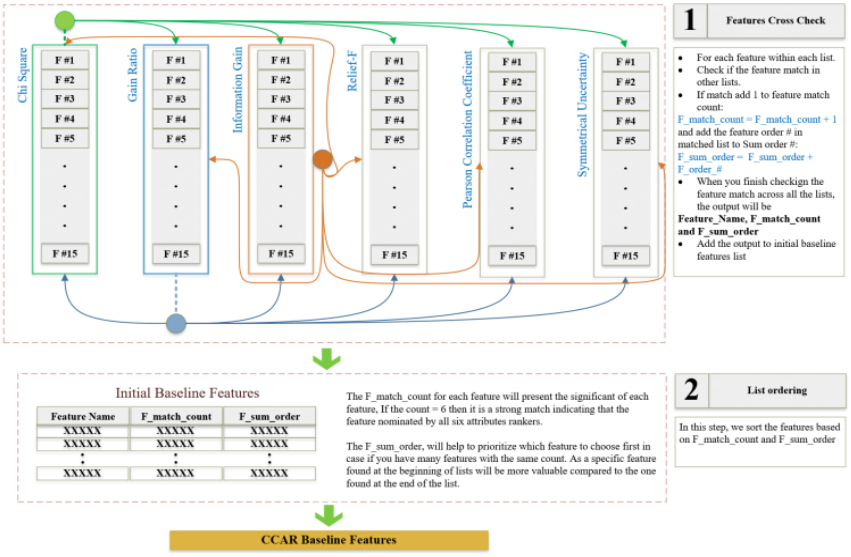

Figure 6. Overview of the CCAR Framework

Location	Chi Square	Gain Ratio	Information Gain	Pearson Correlation	Relief-F	Symmetrical Uncertainty
1	PctExtHyperlinks	FrequentDomainNameMismatch	PctExtHyperlinks	PctExtNullSelfRedirectHyperlinksRT	PctExtNullSelfRedirectHyperlinksRT	PctExtHyperlinks
2	PctExtResourceUrls	PctExtNullSelfRedirectHyperlinksRT	PctExtResourceUrls	FrequentDomainNameMismatch	InsecureForms	PctExtNullSelfRedirectHyperlinksRT
3	PctExtNullSelfRedirectHyperlinksRT	PctExtHyperlinks	PctExtNullSelfRedirectHyperlinksRT	NumDash	FrequentDomainNameMismatch	PctNullSelfRedirectHyperlinks
4	PctNullSelfRedirectHyperlinks	SubmitInfoToEmail	PctNullSelfRedirectHyperlinks	SubmitInfoToEmail	PctNullSelfRedirectHyperlinks	FrequentDomainNameMismatch
5	NumNumericChars	PctNullSelfRedirectHyperlinks	NumNumericChars	PctNullSelfRedirectHyperlinks	ExtMetaScriptLinkRT	SubmitInfoToEmail
6	ExtMetaScriptLinkRT	IpAddress	FrequentDomainNameMismatch	InsecureForms	IframeOrFrame	PctExtResourceUrls

Figure 7. Example of the CCAR Framework

Features obtained from the cross correlation match, will be sorted according to highest F_match_count values and lowest F_sum_order values, where highest F_match_count values indicates the significant of a particular feature. The highest value in this case will be six which means the feature found in all the six lists, while lowest F_sum_order indicates that the feature often found close to the top of the list, hence considered more significant compared to features often found at middle or end of the list.

In the figure 7 an illustrated example of how the CCAR works. Let assume that six attributes are nominated by each attribute ranker and location column is the order of each feature within each list. For instance, the feature "PctExtHyperlinks" at order number one in Chi Square list, while the same feature at order number three in Gain Ratio list. So, if we continue the match of same feature following the same approach, we should obtain the following information:

The PctExtHyperlinks feature as highlighted in light blue found four matches in Chi Square, Gain Ratio, Information Gain and Symmetrical Uncertainty lists which means the F_match_count = four in this case, while F_sum_order = six computed based on the found feature location in each list. In this case the feature PctExtHyperlinks found at location number one in Chi Square, Information Gain and Symmetrical Uncertainty, while found at location three in Gain Ratio (1+3+1+1 = 6).

Table 2. List of Features Obtained from the Example

Feature	F_match_count	F_sum_order
PctExtNullSelfRedirectHyperlinksRT	6	12
PctNullSelfRedirectHyperlinks	6	25
FrequentDomainNameMismatch	5	16
PctExtHyperlinks	4	6
PctExtResourceUrls	3	10
SubmitInfoToEmail	3	13
InsecureForms	2	8
NumNumericChars	2	10
ExtMetaScriptLinkRT	2	11
NumDash	1	3
IpAddress	1	6
IframeOrFrame	1	6

As shown in the Table 2, the list of features obtained after applying the CCAR. By looking at the list it is a lot easier to identify the baseline features based on the highest F_match_count and lowest F_sum_order. If we assume that our target is to nominate five baseline features, then the first five features will be the ones to choose. Furthermore, if we were to choose between PctExtHyperlinks and PctExtResourceUrls as fifth feature, because both have the same F_match_count equal to three, the F_sum_order will help to which feature to nominate in this case by nominating the feature with lowest F_sum_order.

E. PHASE 3: APPLICATION OF MACHINE LEARNING CLASSIFIERS

This phase is the final phase in the proposed methodology where the produced baseline features from the cross-correlation phase will be used as main baseline dataset for performance evaluation. Leveraging on Weka tool a set for nominated machine learning classifiers algorithms will be used to evaluate the new optimized baseline dataset performance. In this experiment, no other settings customization has been applied in Weka rather than using cross validation (Folds-10) for test options. The other settings are kept the same as the original settings customized by HEFS. The same six classifiers applied in HEFS will be applied in this phase namely Naïve Bayes, C4.5, JRip, SVM, PART and Random Forest in order to have a comparable performance.

F. PERFORMANCE EVALUATION CRITERIA

In this project, confusion matrix will be used to measure the performance of the classification model. Four measuring criteria which are recall, precision, F-Measure and accuracy. These measuring criteria incorporate four values to calculate the performance of each measuring criteria. The values are True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN) and each of these values will be explained below.

- True positive (TP): cases where the classifier predicted TRUE (It is phishing URL), and correct classification was TRUE (the URL is phishing).
- True negative (TN): cases where the model predicted FALSE (not phishing URL), and correct classification was FALSE (the URL not phishing).
- False positive (FP): cases where the classifier predicted TRUE, but the URL not phishing, and it is a legitimate URL.
- False negative (FN): cases where the classifier predicted FALSE (URL is legitimate), but actually the URL was phishing.

The confusion matrix is a well-known performance measurement in machine learning classification based on the predication verses actual as shown in the Figure 8:

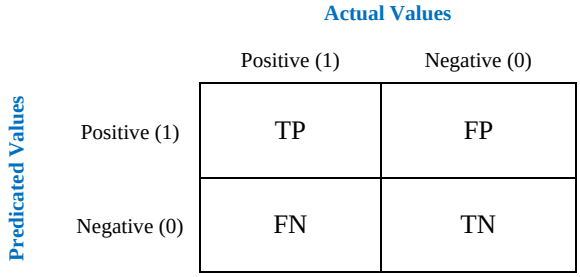

Figure 8. Illustration of Confusion Matrix

1) Accuracy

Accuracy = (TP + TN)/(TP + TN + FP + FN)

It tells how much classification is done correctly. TP and TN together are the correct number of classifications done by the classifier. It does not consider positives and negatives separately, and hence other measures are also required for the analysis other than accuracy.

2) Precision

Precision = TP/(TP + FP)

It indicates how many instances, which are classified as positive, are relevant. High precision is desirable because high relevancy in detecting positives is desired; i.e., less FP is desired.

3) Recall

Recall = TP/(TP + FN)

It is also called a TP rate and is an indication of how good a system can detect positives.

4) F-Measure

F-Measure = 2*(Precision*Recall)/(Precision + Recall)

Since a high Precision and Recall is desired, hence high F-Measure is also desired.

4. Results and Discussion

The experiment conducted on the following hardware specifications:

- Processor: Intel Core I5-2430M 2.4Hz
- Memory: 8 GB DDR3
- Hard Disk: 50 GB.

This section we will shed some light on the redeployment results and discuss the findings, followed by the evaluation of the proposed methodology and the obtained results. Our proposed methodology called cross-correlation helps to identify the optimal features sub-set by excluding the least significant features. Our proposed method.is divided into three stages starting with applying the attributes ranker evaluators on the full dataset to generate a set of most significant features. After that we use cross-correlation function to generate the optimized baseline features and use these features to evaluate and compare our results with the HEFS techniques on different types of machine learning classifiers. Phishing attacks has been proven to be the most effective for hackers counting on the weakest line of defence in the security system which is the end-users. Hence our proposed method aiming to address and contribute to the ongoing efforts to determine the phishing attacks as accurate as possible.

The goal of this experiment is to evaluate the effectiveness of the proposed baseline features in term of accuracy performance against six different machine learning classifiers algorithms namely Random Forest, C4.5, JRip, PART, Na ïve Bayes and SVM. This is done by applying both full baseline features and proposed baseline features in all the aforementioned machine learning classifiers.

Table 3. Performance of Baseline Features vs Full Features

NO	Classifier	Feature set	# of features	Accuracy (%)
1	Random Forest	Full	48	98.37
2	Random Forest	Baseline	10	97.37
3	C4.5	Full	48	97.31
4	C4.5	Baseline	10	96.79
5	JRip	Full	48	97.30
6	JRip	Baseline	10	96.21
7	PART	Full	48	97.60
8	PART	Baseline	10	96.39
9	SVM	Full	48	93.87
10	SVM	Baseline	10	89.55
11	Na ïve Bayes	Full	48	85.15
12	Na ïve Bayes	Baseline	10	81.85

As can be seen in Table 3, we can observe that the proposed technique able to maintain a very good accuracy when compared to full features set with average of less.-1% lower for Random Forest, C4.5, JRip and PART classifiers, while both SVM and Na ïve Bayes classifiers with average of less.-3.5%.

A direct comparison between our proposed cross-correlation baseline performance and the base work performance by evaluating the primarily accuracy performance while precision, recall and f-measure will be included as secondary criteria to measure the performance. First, we will start by presenting the HEFS's baseline features result, followed by our proposed cross-correlation baseline features results, then we present the comparison of accuracy performance between HEFS and cross-correlation.

Table 4. HEFS's Baseline Features Performance

Classifier	Accuracy	Precision	Recall	F-measure
Random Forest	96.34%	96.40%	96.30%	96.30%
C4.5	95.80%	95.80%	95.80%	95.80%
JRip	95.45%	95.50%	95.50%	95.40%
PART	95.23%	95.20%	95.20%	95.20%
SVM	86.71%	86.90%	86.70%	86.70%
Na ïve Bayes	83.25%	84.90%	83.30%	83.10%

As shown in Table 4, we can observe that Random Forecast outperformed the other classifiers which is consistent with the results reported in HEFS. However, the performance accuracy of Random Forest seemed higher where the obtained results show 96.34% versus 94.6% as reported in HEFS.

Table 5. The Proposed Baseline Features Performance

Classifier	Accuracy	Precision	Recall	F-measure
Random Forest	97.37%	97.40%	97.40%	97.40%
C4.5	96.79%	96.80%	96.80%	96.80%
JRip	96.21%	96.20%	96.20%	96.20%
PART	96.39%	96.40%	96.40%	96.40%
SVM	89.55%	89.60%	89.60%	89.50%
Na ïve Bayes	81.85%	84.00%	81.90%	81.60%

In the Table 5, we can see that the features obtained from the proposed technique consistently outperform in all the metrics accuracy, precision, recall and F-measure expect in Na ïve Bayes classifier mainly due to low recall and F-measure.

Table 6. Accuracy Performance Cross-Correlation vs HEFS

Calssifier	Cross-Correlation	HEFS	Variation
Random Forest	97.37%	96.34%	1.03%
C4.5	96.79%	95.80%	0.99%
PART	96.39%	95.23%	1.16%
JRip	96.21%	95.45%	0.76%
SVM	89.55%	86.71%	2.84%
Na ïve Bayes	81.85%	83.25%	-1.40%

From the Table 6, we have put side by side the accuracy results from both HEFS and the proposed cross-correlation model including the variation to have a quick comparison. From the variation data it is very clear that the features obtained from the proposed cross-correlation model are consistently achieving higher accuracy in all the classifiers apart from Naïve Bayes where the HEFS's features perform better.

Most importantly when we look at the main evaluation criteria which is the Random Forest classifier performance with the baseline features, we can clearly validate that effectiveness of our optimized cross-correlation baseline features achieving 1.03% improvement in accuracy.

Furthermore, As illustrated in the Figure 9 below shows the accuracy evaluation where the accuracy performance of the anchor's baseline using Random Forest classifier achieved the highest performance 94.60% among the other classifiers, but when we re-implementation the work using the same HEFS's baseline features and the same dataset we found that the accuracy actually was 96.34% which is higher than 94.60%.

We could not find a logical explanation despite the fact that we have exactly re-implemented the work using the same settings and configuration and this is actually introduced more challenges to our proposed baseline features by raising the bar of our benchmark from 94.60% to 96.34%, nevertheless our proposed cross-correlation methodology and the produced baseline features outperformed the previous work where the accuracy obtained is 97.37%.

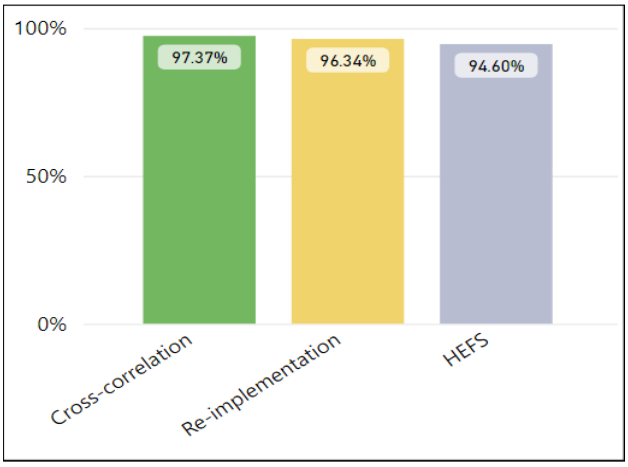

Figure 9. Evolution of Accuracy Performance of HEFS, Re-implementation and Cross-Correlation

5. Conclusions

Phishing attacks and social engineering tactics are getting more sophisticated, more targeted, and more advanced. Despite all the efforts towards the anti-phishing techniques, there are many phishing attacks instances gone unreported. This is raising a major concern since the human factor plays a significant role in the anti-phishing defence system, hence leveraging on technology became prominently a must to automate the phishing detection with highest accuracy possible and shift the overburden load from end users to technology.

The main objective of this project is to improve the accuracy of phishing attack detection by reducing the false positive rate. In order to achieve that, we proposed a feature selection technique called cross-correlation attribute ranker (CCAR) by leveraging the existing attributes rankers' algorithms such as Information Gain, Gain Ratio, Chi-Square and few more. To evaluate the effectiveness of the proposed technique, we have measured and compared our results with the benchmark scheme called HEFS. Results show that our proposed technique clearly suppress almost all the accuracy obtained by the HEFS scheme. Although, several results obtained from our proposed work are not as accurate with the HEFS scheme. The obtained accuracy results are acceptable considering the significant reduction of the number of features being used in our evaluation.

6. Future Work

About future work, writing code for the proposed cross-correlation model CCAR can improve and automate the process and potentially to explore more attributes rankers on top on the six rankers used in our project. Also, exposing the baseline features to other datasets can also help to

validate the effectiveness of the proposed model and baseline dataset. In addition, exploring more machine learning classifiers can be another area for future work direction. Furthermore, looking at more optimizing baseline features without compromising the other performances could also be one of the promising future work directions.

REFERENCES

[1] Basnet, R. B., Sung, A. H., & Liu, Q. (2012). Feature selection for improved phishing detection. In International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems (pp. 252-261). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-31087-4_27.

[2] Buber, E., Dırı, B., & Sahingoz, O. K. (2017). Detecting phishing attacks from URL by using NLP techniques. In 2017 International conference on computer science and Engineering (UBMK) (pp. 337-342). IEEE. https://doi.org/10.1109/UBMK.2017.8093406.

[3] Chiew, K. L., Chang, E. H., & Tiong, W. K. (2015). Utilisation of website logo for phishing detection. Computers & Security, 54, 16-26. https://doi.org/10.1016/j.cose.2015.07.006.

[4] Chiew, K. L., Tan, C. L., Wong, K., Yong, K. S., & Tiong, W. K. (2019). A new hybrid ensemble feature selection framework for machine learning-based phishing detection system. Information Sciences, 484, 153-166. https://doi.org/10.1016/j.ins.2019.01.064.

[5] Feng, F., Zhou, Q., Shen, Z., Yang, X., Han, L., & Wang, J. (2018). The application of a novel neural network in the detection of phishing websites. Journal of Ambient Intelligence and Humanized Computing, 1-15. https://doi.org/10.1007/s12652-018-0786-3.

[6] Govil, N., Agarwal, K., Bansal, A., & Varshney, A. (2020). A Machine Learning based Spam Detection Mechanism. In 2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC) (pp. 954-957). IEEE. https://doi.org/0.1109/ICCMC48092.2020.ICCMC-000177.

[7] Gualberto, E. S., De Sousa, R. T., Thiago, P. D. B., Da Costa, J. P. C., & Duque, C. G. (2020). From feature engineering and topics models to enhanced prediction rates in phishing detection. Ieee Access, 8, 76368-76385. https://doi.org/10.1109/ACCESS.2020.2989126.

[8] Jain, A. K., & Gupta, B. B. (2018). Towards detection of phishing websites on client-side using machine learning based approach. Telecommunication Systems, 68(4), 687-700. https://doi.org/10.1007/s11235-017-0414-0.

[9] Jeeva, S. C., & Rajsingh, E. B. (2016). Intelligent phishing url detection using association rule mining. Human-centric Computing and Information Sciences, 6(1), 1-19. https://doi.org/10.1186/s13673-016-0064-3.

[10] Khonji, M., Jones, A., & Iraqi, Y. (2013). An empirical evaluation for feature selection methods in phishing email classification. International Journal of Computer Systems Science & Engineering, 28(1), 37-51.

[11] Le, A., Markopoulou, A., & Faloutsos, M. (2011, April). Phishdef: Url names say it all. In 2011 Proceedings IEEE INFOCOM (pp. 191-195). IEEE. https://doi.org/10.1109/INFCOM.2011.5934995.

[12] Li, J. H., & Wang, S. D. (2017, November). PhishBox: an approach for phishing validation and detection. In 2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech) (pp. 557-564). IEEE. https://doi.org/10.1109/DASC-PICom-DataCom-CyberSciTec.2017.101.

[13] Machado, L., & Gadge, J. (2017). Phishing Sites Detection Based on C4. 5 Decision Tree Algorithm. In 2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA) (pp. 1-5). IEEE. https://doi.org/10.1109/ICCUBEA.2017.8463818.

[14] Moghimi, M., & Varjani, A. Y. (2016). New rule-based phishing detection method. Expert systems with applications, 53, 231-242. https://doi.org/10.1016/j.eswa.2016.01.028.

[15] Mohammad, R. M., Thabtah, F., & McCluskey, L. (2014). Predicting phishing websites based on self-structuring neural network. Neural Computing and Applications, 25(2), 443-458. https://doi.org/10.1007/s00521-013-1490-z.

[16] Orunsolu, A. A., Sodiya, A. S., & Akinwale, A. T. (2019). 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.

[17] Patil, S., & Dhage, S. (2019). A methodical overview on phishing detection along with an organized way to construct an anti-phishing framework. In 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS) (pp. 588-593). IEEE. https://doi.org/10.1109/ICACCS.2019.8728356.

[18] Peng, T., Harris, I., & Sawa, Y. (2018). Detecting phishing attacks using natural language processing and machine learning. In 2018 ieee 12th international conference on semantic computing (icsc) (pp. 300-301). IEEE. https://doi.org/10.1109/ICSC.2018.00056.

[19] Qabajeh, I., & Thabtah, F. (2014). An experimental study for assessing email classification attributes using feature selection methods. In 2014 3rd International Conference on Advanced Computer Science Applications and Technologies (pp. 125-132). IEEE. https://doi.org/10.1109/ACSAT.2014.29.

[20] Rashid, J., Mahmood, T., Nisar, M. W., & Nazir, T. (2020). Phishing Detection Using Machine Learning Technique. In 2020 First International Conference of Smart Systems and Emerging Technologies (SMARTTECH) (pp. 43-46). IEEE. https://doi.org/10.1109/SMART-TECH49988.2020.00026.

[21] Sahingoz, O. K., Buber, E., Demir, O., & Diri, B. (2019). Machine learning based phishing detection from URLs. Expert Systems with Applications, 117, 345-357. https://doi.org/10.1016/j.eswa.2018.09.029.

[22] Sanglerdsinlapachai, N., & Rungsawang, A. (2010). Using domain top-page similarity feature in machine learning-based web phishing detection. In 2010 Third International Conference on Knowledge Discovery and Data Mining (pp. 187-190). IEEE. https://doi.org/10.1109/WKDD.2010.108.

[23] Shyni, C. E., Sundar, A. D., & Ebby, G. E. (2018). Phishing Detection in Websites using Parse Tree Validation. In 2018 Recent Advances on Engineering, Technology and Computational Sciences (RAETCS) (pp. 1-4). IEEE. https://doi.org/10.1109/RAETCS.2018.8443961.

[24] Singh, B., Kushwaha, N., & Vyas, O. P. (2014). A feature subset selection technique for high dimensional data using symmetric uncertainty. Journal of Data Analysis and Information Processing, 2(04), 95. https://doi.org/10.4236/jdaip.2014.24012.

[25] Smadi, S., Aslam, N., & Zhang, L. (2018). Detection of online phishing email using dynamic evolving neural network based on reinforcement learning. Decision Support Systems, 107, 88-102. https://doi.org/10.1016/j.dss.2018.01.001.

[26] Sonowal, G., & Kuppusamy, K. S. (2020). PhiDMA–A phishing detection model with multi-filter approach. Journal of King Saud University-Computer and Information Sciences, 32(1), 99-112. https://doi.org/10.1016/j.jksuci.2017.07.005.

[27] Subasi, A., & Kremic, E. (2020). Comparison of adaboost with multiboosting for phishing website detection. Procedia Computer Science, 168, 272-278. https://doi.org/10.1016/j.procs.2020.02.251.

[28] Tan, C. L., & Chiew, K. L. (2014). Phishing website detection using URL-assisted brand name weighting system. In 2014 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS) (pp. 054-059). IEEE. https://doi.org/10.1109/ISPACS.2014.7024424.

[29] Toolan, F., & Carthy, J. (2010). Feature selection for spam and phishing detection. In 2010 eCrime Researchers Summit (pp. 1-12). IEEE. https://doi.org/10.1109/ecrime.2010.5706696.

[30] Tuteja, S. K., & Bogiri, N. (2016). Email Spam filtering using BPNN classification algorithm. In 2016 International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT) (pp. 915-919). IEEE. https://doi.org/10.1109/ICACDOT.2016.7877720.

[31] Thabtah, F., & Abdelhamid, N. (2016). Deriving correlated sets of website features for phishing detection: a computational intelligence approach. Journal of Information & Knowledge Management, 15(04), 1650042. https://doi.org/10.1142/S0219649216500428.

[32] Varshney, G., Misra, M., & Atrey, P. K. (2016). A phish detector using lightweight search features. Computers & Security, 62, 213-228. https://doi.org/10.1016/j.cose.2016.08.003.

Copyright © 2026 The Author(s). Published by Scientific & Academic Publishing
This work is licensed under the Creative Commons Attribution International License (CC BY). http://creativecommons.org/licenses/by/4.0/