© Universiti Tun Hussein Onn Malaysia Publisher's Office

JSCDM

Journal homepage: http://penerbit.uthm.edu.my/ojs/index.php/jscdm

Journal of Soft Computing and Data Mining

Feature Selection to Enhance Phishing Website Detection Based On URL Using Machine Learning Techniques

Liyana Mat Rani[1,2], Cik Feresa Mohd Foozy[1*] Siti Noor Baini Mustafa[3]

[1]Faculty of Computer Science and Information Technology,
Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, Johor, MALAYSIA

[2]Department of Information Technology and Communication,
Politeknik Sultan Mizan Zainal Abidin, KM 8, Jalan Paka, 23000 Dungun, Terengganu, MALAYSIA

[3]Book Hack Enterprise,
Bandar Baru Nilai,71800 Nilai, Negeri Sembilan, MALAYSIA

DOI: https://doi.org/10.30880/jscdm.2023.04.01.003
Received 02 February 2023; Accepted 12 April 2023; Available online 25 May 2023

Abstract: The detection of phishing websites based on machine learning has gained much attention due to its ability to detect newly generated phishing URLs. To detect phishing websites, most techniques combine URLs, web page content, and external features. However, the content of the web page and external features are time-consuming, require large computing power, and are not suitable for resource-constrained devices. To overcome this problem, this study applies feature selection techniques based on the URL to improve the detection process. The methodology for this study consists of seven stages, including data preparation, preprocessing, splitting the dataset into training and validation, feature selection, 10-fold cross-validation, validating the model, and finally performance evaluation. Two public datasets were used to validate the method. TreeSHAP and Information Gain were used to rank features and select the top 10, 15, and 20. These features are fed into three machine learning classifiers which are Naïve Bayes, Random Forest, and XGBoost. Their performance is evaluated based on accuracy, precision, and recall. As a result, the features ranked by TreeSHAP contributed most to improving detection accuracy. The highest accuracy of 98.59 percent was achieved by XGBoost for the first dataset with 15 features. For the second dataset, the highest accuracy is 90.21 percent using 20 features and Random Forest. As for Naïve Bayes, the highest accuracy recorded is 98.49 percent using the first dataset.

Keywords: Machine learning, phishing detection, feature selection, information gain, TreeSHAP

1. Introduction

Phishing is a social engineering attack that aims to steal a user's identity data and financial account credentials [1]. Attackers exploit the lack of cybersecurity awareness among users and the insecure Internet protocol to trick users into visiting phishing websites. The current problem in phishing website detection is the inability of the list-based and visual similarity technique to detect zero-day phishing attacks due to the nature of the phishing websites, which only appear for a short time [2], [3]. Recent phishing techniques use hybrid features, which combine URL, content, and external-based [4]-[6]. However, web page content and external-based features are time-consuming [7], [8]. Many past works of literature used a variety of features but did not include any information about feature selection [9], [10], which is crucial since using many features requires a lot of computing power and is not suitable for resource-constrained devices such as mobile phones [11]. Therefore, this study aims to employ a feature selection technique based on URLs using machine learning to improve the detection of phishing websites based on URLs.

The objectives of this research are (1) to study feature profiles for phishing website detection based on URL, (2) to develop a feature selection model for phishing website detection based on URL using machine learning techniques, (3)

*Corresponding author: feresa@uthm.edu.my

to validate the website phishing detection model in terms of accuracy, precision, and recall. This study will make three contributions. The first is feature profiles that can be applied to phishing website detection. Second, the development of a feature selection model to detect phishing websites using ML techniques. Third, the evaluation of the proposed model's performance, compared with state-of-the-art techniques with respect to accuracy and efficiency.

The body of this article is organized as follows. Section 2 discusses the review of related works. Section 3 discusses the research methodology. Section 4 presents the experimental results, discussion, and comparative analysis with existing methods. Lastly, Section 5 draws conclusions and suggests future works.

2. Related Works

An overview of current methods for detecting phishing attacks is provided in this section.

2.1 Phishing

Phishing attacks are used to steal internet users' sensitive data, such as passwords, social security numbers, and credit card details. The objectives are achieved by exploiting people with very limited digital or cyber security awareness or who are poorly trained [12]. Phishing is a serious threat that causes billions of dollars' worth of losses for businesses and individuals [13].

2.2 Phishing Technique

Phishing attacks employ several techniques. These include email, spear, whaling, voice, mobile application, and website phishing. Phishing emails use fake emails to trick victims into visiting malicious websites. Spear phishing also uses email but targets a specific individual. Whaling involves phishing emails targeted at high-ranking employees. Voice phishing attacks involve an attacker posing as a government official, tax official, bank representative, etc. Smartphone apps can also be used for phishing attacks. There is a risk that downloaded applications contain malware. Lastly, website phishing involves creating a fake website that mimics a legitimate website. A fake website could have forms or buttons that lead the user to an attacker's trap. Most transactions take place online. The most common way to become a victim of these attackers is through a website [14].

2.3 Phishing Website Detection Technique

There are three techniques for detecting phishing websites which are list-based, visual similarity and machine learning [2]. The list-based technique maintained the blocked URL in a database. When users initiate a web page, the browser searches the blacklist database for the page and alerts them if it is found. In the visual similarity-based technique, attackers create a webpage that resembles a legitimate website. Machine Learning (ML) uses features extracted from URLs, source code, website traffic, domains, etc. These features are then used to create a dataset. The dataset is then pre-processed and fed into the ML algorithm. Recently, many authors proposed ML techniques for phishing website detection [11], [12] because of their capability to detect zero-day attacks.

2.4 Features of Phishing Website

There are three groups of features used in the ML technique which are URL-based, content-based, and external-based. Many phishing techniques use hybrid features, which combine the URL, content, and external-based features [7]. The URL-based features depend on the characteristics of the URL, such as the use of IP address, blacklisted words, use of HTTPS, length of URL, etc. The content-based approach requires an in-depth analysis of the pages content [1]. Some of the most common tricks followed by attackers while designing a phishing website are to disallow users to view the source code of the web page. External-based features rely on third-party services such as search engine indexing, WHOIS, and page rank. The URL-based method is only dependent on the URL characteristics. Amongst all these techniques, URL is the only technique that is not web dependent, its processing time is very low, and the accuracy of phishing URL detection is very high [15].

Several public phishing datasets available are from University of Irvine California (UCI) and ISCX-URL-2016 from University of New Brunswick (UNB). The Mendeley repositories provide several sets of phishing dataset. One of the datasets is contributed by Hannousse & Yahiouche [16] is used in this study.

2.5 Related Work on Phishing Website Detection Using ML Technique

There are many ML techniques to detect phishing website has been applied. UÇAR [17] uses 79 defined features from ISCX-URL2016. They apply Long Short-Term Memory (LSTM) and Convolutional Neural Networks (CNN) to validate the use of deep learning algorithms. This technique achieved the highest accuracy of 93.67 percent using CNN.

Rao et al. [8] proposed features that use the hostname, full URL, Term Frequency-Inverse Document Frequency (TF-IDF), and phishing-hinted words from the suspicious URL to detect phishing websites. A total of three datasets were created, D1, D2, and D3. In the experiment, the RF classifier achieved an accuracy of 94.26 percent on D1.

Basit et al. [18] used ensemble ML methods including Artificial Neural Networks (ANN), K-Nearest Neighbors (KNN), and Decision Trees (C.45) together with RF classifiers to detect phishing websites. Using KNN and RF, the model achieved 97.33 percent accuracy on the UCI dataset.

Geyik et al. [19] utilized the dataset produced by Rao et al. [8]. A total of 15 URL features were extracted from the datasets. The highest accuracy of 83 percent was achieved using RF. Using a balanced and enhanced dataset is expected to improve detection accuracy since the dataset is imbalanced.

Atari & Al-Mousa [20] use the dataset from Hannousse & Yahiouche [16] to discover patterns and relationships not evident in the raw data. The authors evaluate several ML classifiers based on accuracy, recall, and precision. It achieves 97 percent accuracy with XGBoost. Table 1 summarizes the works by previous researchers.

Table 1 - Summary of related work on ML-based detection techniques

Index	References	Feature Types	Dataset	Classification Algorithm	Accuracy (%)
1.	UÇAR [17]	URL	ISCX-URL2016	CNN	93.67
2.	Rao et al. [8]	URL	Own Dataset	RF	94.26
3.	Basit et al. [18]	Hybrid	UCI	KNN & RF	97.33
4.	Geyik et al. [19]	URL	Rao et al . [8] (D3)	RF	83
5.	Atari & Al-Mousa [20]	Hybrid	Hannousse & Yahiouche [16]	XGBoost	97

2.6 Feature Selection Technique

The study applied two feature selection techniques which is Information Gain (IG) and Shapley Additive Explanation (SHAP). IG is a popular filter method used to reduce the dimension of data [21]. SHAP combines optimal credit allocation with local explanations based on classical Shapley values based on a game-theoretic approach. This value shows how much a feature contributes to a change in the model's output.

2.7 Related Work on Feature Selection Technique

Several studies have been conducted on feature selection. Adi et al. [21] selected the 9 most significant features in the UCI dataset using the Gain Ratio, achieving an accuracy of 94.17 percent. As compared to accuracy with all 30 features in the dataset, feature selection has a negative impact on classification accuracy. With 30 features, the Decision Tree classifier achieved an accuracy of 96.73 percent. A decrease in accuracy occurs either because of the characteristics of the data or the selected features. Zaman et al. [22] obtained accuracy of 95.8 percent using the UCI dataset. Classification was performed on the UCI dataset based on four categories. HNB classifier gave the highest accuracy with Address Bar features. With the combination of J48 and HNB, accuracy increased to 96.25 percent after the researchers applied filter feature selection. While Gandotra & Gupta [23] used IG to select the 15 most significant features and achieved 96.3 percent accuracy with RF. Without feature selection, accuracy increases only by 96.52 percent. The model used with reduced features selected by Ig show a comparable performance with the model without any feature selection.

Kasim [24] identified 77 URL-based features from the ISCX-URL-2016 dataset. The number of features was reduced to 20 using Sparse Autoencoder (SAE) and PCA. With LightGBM, 99.6 percent accuracy was achieved. Bu & Kim [25] applied the Genetic Algorithm (GA) as feature selection to improve accuracy and recall. With 15 features, the deep learning model achieved accuracy of 96.85 percent and recall of 95.10 percent. With just nine features from URLs, Gupta et al. [11]improved accuracy and recall. It was done by analyzing several available lexical features proposed by different past researchers. Several algorithms were used to calculate feature importance. Spearman correlation, K-best, and RF are used to determine feature importance. Their accuracy and recall were 99.57 and 99.46 percent, respectively.

Hannousse & Yahiouche [16] developed a phishing dataset that includes 56 URL-based, 24 content-based, and seven external-based features. By using Chi-Square, 73 out of 87 features the model achieved 96.86 percent accuracy. External-based features showed the highest accuracy of 94.09 percent even though the group only contained a few features. URL-based features scored 91.03 percent accurate, followed by content-based features with 89.87 percent. Moedjahedy et al. [26] propose a method combining correlation and Recursive Feature Elimination (RFE) to determine which URL characteristics are useful for identifying phishing websites by gradually decreasing the number of features while maintaining accuracy. The accuracy obtained is 97.06 percent using RF. Table 2 summarizes the feature selection techniques that have been used by past researchers.

Table 2 - Summary of related work on feature selection techniques

Index	References	Feature Selection Technique	Dataset (No of Features)	No. Feature Selected	Classification Algorithm	Accuracy (%)
1.	Adi et al. [21]	Gain Ratio	UCI (30)	9	Decision Tree	9417
2.	Zaman et al. [22]	Filter	UCI (30)	27	HNB+J48	96.25
3.	Gandotra & Gupta [23]	IG	UCI (30)	15	RF	96.3
4.	Kasim [24]	SAE + PCA	ISCX-URL-2016 (77)	20	LightGBM	99.6
5.	Bu & Kim [25]	GA	ISCX-URL-2016 (91)	15	Deep Learning	96.85
6.	Hannousse & Yahiouche [16]	Chi Square	Hannousse & Yahiouche (87)	77	RF	94.09
8.	Moedjahedy et al. [26]	Combines correlation and RFE	Hannousse & Yahiouche (87)	10	RF	97.6
7.	Gupta et al. [11]	Manually Using Spearman Correlation, K-Best, and RF	ISCX-URL-2016	9	RF	99.57

3. Methodology

There are seven phases used in this research which are data preparation, data preprocessing, split data into training and validation, feature selection, 10-fold cross validation, validate data and performance evaluation as shown in Figure 1.

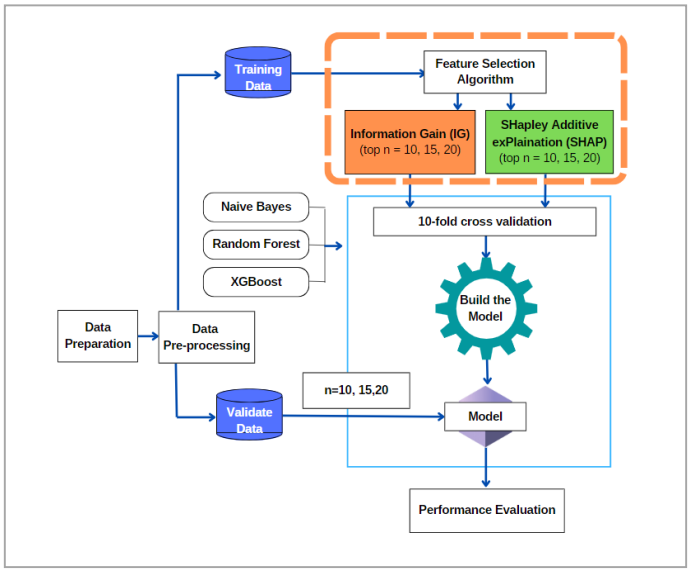

Fig. 1 - Research model

3.1 Data Preparation

Two dataset is used in this study which are:

- Dataset 1: ISCX-URL-2016
- Dataset 2: Hannousse & Yahiouche (2020)

3.2 Data Preprocessing

The dataset contains missing value which will be replaced with mean value. The redundant data will be removed, and the class label of the datasets will be change to ‘0’ for legitimate URL and ‘1’ for phishing URL.

3.3 Split into Training and Validation Data

The dataset is split into training and validation data with ratio of 70:30. The training data is to be used in 10-fold cross validation. The validation data is used to validate the performance with the unseen data to see if the model able to generalize well.

3.4 Feature Selection Algorithm

This study utilizes two feature selection techniques which are IG and SHAP to select the most important features value. The features are ranked from highest to lowest and the top 10,15 and 20 are selected for training the ML model. The calculation of IG is referred to as Mutual Information (MI) between the two random variables. MI between two random variables is a non-negative value, which measures the dependency between the variables. The formulae to calculate IG is given in (1) and (2).

$$Entropy = -\sum P(x) \times log_2 P(x) \quad (1)$$

$$IG\ (X;\ Y) = H(X) - H\ (X \mid Y) \quad (2)$$

There are three methods of using SHAP values which are kernel-based, linear-based, and tree-based. Lundberg et. al. [27] proposed TreeSHAP, a variant of SHAP for tree-based machine learning models which is much faster. The formulae for average of absolute Shapley values per feature across the data is calculated using (3).

$$Ij = \frac{1}{n} \sum_{i=1}^{n} \left| \oint_j^{(i)} \right| \quad (3)$$

3.5 10-Fold Cross Validations

10-fold cross validation divides a data set into 10 subsets. Each time, one subset is used as the test set and the other nine as a training set. A second subset of data will be used as test data, and the remainder as training data. This is repeated 10 times. The average error is calculated across all 10 trials.

3.6 Validate The ML Model

To validate the performance of ML datasets, data that has been separated in the early phase is used. The classification algorithms used are Naïve Bayes, Random Forest, and XGBoost.

The Naïve Bayes algorithm assumes that each feature makes an independent and equal contribution to the outcome. Equation 4 provides a way of calculating the probability of P(y|X) from P(X), P(Y), and P(X|y).

$$P(y|X) = \frac{P(X|y)P(y)}{P(X)} \quad (4)$$

In the RF algorithm, several decision trees are constructed, and predictions are derived from them. Trees are based on predefined attributes selected randomly. Classification is done by majority vote for each tree.

In XGboost, a set of individual models are combined to create a final model. As opposed to Random Forest, which creates all the Decision Trees at once, XGBoost creates decision trees sequentially. All independent variables are weighted or given residual values, which are then fed into the decision tree. The residual values of variables classified wrongly by the tree are increased and fed to a second decision tree. These individual classifiers are then ensembled to create a stronger and more precise model.

3.7 Performance Evaluation

The evaluation metrics used to evaluate the classifier performance used in this research are accuracy, precision, and recall.

- **Accuracy**: Total number of overall correctly classified instances. It is the number of correct classifications of either Phishing or Legitimate URLs out of all URLs in the dataset. The accuracy of the model is calculated using formulae in (5).

$$\frac{TP + TN}{TP + TN + FN + FP} \quad (5)$$

- **Precision**: Total number of URLs detected as phishing out of total phishing URLs. The precision of the model is calculated using formulae in (6).

$$\frac{TP}{TP + FP} \quad (6)$$

- **Recall**: Total number of legitimate URLs classified as legitimate and phishing URLs classified as phishing. The recall value of the model is calculated using formulae in (7).

$$\frac{TP}{TP + FN} \quad (7)$$

4. Result and Discussion

This section describes the findings of this study.

4.1 Implementation Tools

Experiments were conducted on a machine with a 2.50 GHz core i5 processor, 12 GB of RAM, and a 128 MB graphics card. To implement the proposed approach, Jupyter Notebook 6.4.8 with Python 3.9 was used.

4.2 Dataset Preprocessing

In both datasets, class labels are first converted to numerical values during preprocessing. Class 'phishing' is converted to '1', while 'benign' or 'legitimate' is assigned to '0'. The first dataset, ISCX-URL-2016 contains many null values. These values were replaced with the mean. Due to the potential for bias in performance estimates, duplicate data is removed. This research only used URL-based features from Dataset 2. The dataset is almost ready to be fed into ML algorithms. The cleaning involved removing the URL attributes, changing the categorical values in the class label, and removing duplicate values.

4.3 Split Dataset into Training and Validation

To evaluate the performance of the ML models, the dataset was split into training and validation with a 70:30 ratio.

4.4 Feature Selection

To use IG, the Mutual Information (MI) is calculated using the sklearn library. While the TreeSHAP value is obtained using the SHAP library with XGBoost as the classifier. The top 20 features are rank according to result obtained for both datasets. Table 3, 4, 5 and 6 show the top 20 feature ranked by IG and TreeSHAP for the first and the second datasets.

Table 3 - Top 20 feature ranked by IG for dataset 1

Ranking	Feature Selected	MI Value	Ranking	Feature Selected	MI Value
1	Entropy_Domain	130.9588	11	Entropy_Filename	66.21004
2	domainUrlRatio	112.7052	12	delimeter_path	61.81792
3	pathDomainRatio	108.83	13	ArgUrlRatio	57.11311
4	LongestPathTokenLength	101.9097	14	NumberRate_URL	50.0853
5	CharacterContinuityRate	100.5995	15	NumberRate_FileName	44.67871
6	argDomanRatio	95.19503	16	Extension_LetterCount	43.11572
7	host_letter_count	78.2705	17	Entropy_DirectoryName	41.61881
8	NumberofDotsinURL	75.16174	18	Filename_LetterCount	40.74079
9	SymbolCount_Domain	70.81768	19	URL_DigitCount	39.03056
10	argPathRatio	68.78684	20	SubDirectoryLongestWordLength	35.5193

Table 4 - Top 20 feature ranked by TreeSHAP for dataset 1

Ranking	Feature Selected	MI Value	Ranking	Feature Selected	MI Value
1	domain_token_count	0.569355	11	Extension_DigitCount	0.02837
2	urlLen	0.338641	12	Directory_DigitCount	0.027726
3	domainlength	0.224255	13	SymbolCount_Directoryname	0.027631

4	longdomaintokenlen	0.112515	14	Filename_LetterCount	0.02341
5	LongestPathTokenLength	0.105443	15	Extension_LetterCount	0.022915
6	ldl_url	0.06954	16	fileNameLen	0.011804
7	NumberofDotsinURL	0.061876	17	ArgUrlRatio	0.010323
8	Directory_LetterCount	0.052387	18	ldl_filename	0.009837
9	Entropy_Domain	0.048975	19	domainUrlRatio	0.009393
10	CharacterContinuityRate	0.033213	20	pathLength	0.009156

Table 5 - Top 20 feature ranked by IG for dataset 2

Ranking	Feature Selected	MI Value	Ranking	Feature Selected	MI Value
1	longest_word_path	29.4063	11	tld_in_subdomain	6.927197
2	phish_hints	22.00435	12	shortest_word_path	6.617204
3	longest_words_raw	21.67551	13	longest_word_host	6.342246
4	avg_word_host	21.03599	14	length_words_raw	6.007867
5	avg_word_path	17.82838	15	prefix_suffix	5.749852
6	shortest_word_host	17.04112	16	statistical_report	3.747511
7	avg_words_raw	16.71325	17	domain_in_brand	3.077857
8	char_repeat	16.54574	18	abnormal_subdomain	2.529751
9	shortest_words_raw	8.675617	19	suspecious_tld	2.130794
10	nb_subdomains	7.026079	20	nb_redirection	1.475893

Table 6 - Top 20 feature ranked by TreeSHAP for dataset 1

Ranking	Feature Selected	MI Value	Ranking	Feature Selected	MI Value
1	nb_www	0.447791	11	nb_underscore	0.054381
2	phish_hints	0.263688	12	ratio_digits_url	0.04553
3	longest_word_path	0.190513	13	https_token	0.029032
4	nb_hyphens	0.143736	14	nb_redirection	0.028444
5	domain_in_brand	0.118542	15	shortest_word_path	0.027896
6	nb_dots	0.091188	16	avg_word_host	0.02207
7	ratio_digits_host	0.090072	17	shortening_service	0.021714
8	length_words_raw	0.065985	18	longest_words_raw	0.021707
9	nb_slash	0.06092	19	nb_qm	0.017272
10	length_hostname	0.059405	20	avg_word_path	0.015332

4.5 Result

The performance of the ML model is evaluated with Naïve Bayes, RF, and XGBoost on incremental feature subsets of 10,15,20 and all features. The feature subsets were obtained from IG and TreeSHAP. A training dataset is used to train the dataset, and then an unseen test dataset is used to validate its accuracy. Performance of three ML classifiers is shown in Tables 7 and 8. For the first dataset, the overall performance of the classifiers exceeded 90 percent in accuracy, precision, and recall. All the classifiers were best performed with features selected using TreeSHAP except for Naïve Bayes. With 15 features selected by TreeSHAP and classifier XGBoost, the highest performance was obtained with accuracy, precision, and recall values of 98.561, 0.9556, and 0.9823.

Table 7 - Performance results of ML models using dataset 1

Algorithm	Accuracy	Precision	Recall
Naïve Bayes	**82.932**	**0.9671**	**0.6791**
Naïve Bayes+IG+Top10	84.866	0.9036	0.7779
Naïve Bayes+IG+Top15	91.14	0.9114	0.9098
Naïve Bayes+IG+Top20	90.825	0.8999	0.917
Naïve Bayes+TreeSHAP+Top10	83.719	0.8796	0.7783

Naïve Bayes+TreeSHAP+Top15	84.169	0.8824	0.7856
Naïve Bayes+TreeSHAP+Top20	85.901	0.9055	0.7992
Random Forest	**98.381**	**0.9877**	**0.9796**
Random Forest +IG+Top10	97.819	0.9831	0.9728
Random Forest +IG+Top15	98.313	0.9859	0.9801
Random Forest +IG+Top20	98.291	0.9854	0.98
Random Forest +TreeSHAP+Top10	98.156	0.984	0.9787
Random Forest +TreeSHAP+Top15	98.493	0.9881	0.9814
Random Forest +TreeSHAP+Top20	98.358	0.9881	0.9787
XGBoost	**98.583**	**0.9899**	**0.9814**
XGBoost +IG+Top10	97.954	0.9844	0.9742
XGBoost +IG+Top15	98.358	0.9863	0.9805
XGBoost +IG+Top20	98.336	0.9872	0.9791
XGBoost +TreeSHAP+Top10	98.313	0.9863	0.9796
XGBoost +TreeSHAP+Top15	98.561	0.9886	0.9823
XGBoost +TreeSHAP+Top20	98.561	0.9899	0.981

Based on Table 8, for the second dataset, the performance of RF and XGBoost exceeds 90 percent with 20 selected features. Naive Bayes is underperforming, with accuracy and recall of 76.99 and 0.61, respectively. TreeSHAP selects features that show the highest performance for all classifiers. The highest accuracy was obtained using RF with accuracy, precision, and recall of 90.247 percent, 0.8961, and 0.9191.

Table 8 - Performance results of ML models using dataset 2

Algorithm	Accuracy	Precision	Recall
Naïve Bayes	**69.639**	**0.9367**	**0.4464**
Naïve Bayes+IG+Top10	67.769	0.8874	0.4355
Naïve Bayes+IG+Top15	70.407	0.9008	0.4843
Naïve Bayes+IG+Top20	71.977	0.9174	0.5067
Naïve Bayes+TreeSHAP+Top10	77.288	0.9261	0.6121
Naïve Bayes+TreeSHAP+Top15	76.987	0.9214	0.6095
Naïve Bayes+TreeSHAP+Top20	76.52	0.9178	0.6024
Random Forest	**90.314**	**0.8952**	**0.9216**
Random Forest +IG+Top10	84.335	0.8461	0.8542
Random Forest +IG+Top15	85.404	0.8522	0.8703
Random Forest +IG+Top20	86.64	0.8678	0.8767
Random Forest +TreeSHAP+Top10	86.774	0.8617	0.8882
Random Forest +TreeSHAP+Top15	89.212	0.8881	0.9069
Random Forest +TreeSHAP+Top20	90.281	0.8996	0.9152
XGBoost	**91.049**	**0.9051**	**0.9249**
XGBoost +IG+Top10	83.4	0.8428	0.8369
XGBoost +IG+Top15	85.204	0.8566	0.8593
XGBoost +IG+Top20	85.872	0.8668	0.8606
XGBoost +TreeSHAP+Top10	87.174	0.87	0.8857
XGBoost +TreeSHAP+Top15	89.546	0.8873	0.9152
XGBoost +TreeSHAP+Top20	90.247	0.8961	0.9191

4.6 Comparative Analysis

This section analyzes the results of using IG and TreeSHAP as feature selection algorithms, along with three classifiers, Naïve Bayes, RF, and XGBoost. Figure 2 shows the performance of IG and TreeSHAP for the first dataset. As shown in Figure 2(a), IG obtained the best accuracy with 15 features. XGBoost outperformed other classifiers with 98.358 percent accuracy using 15 features selected using IG. Adding 20 features reduces accuracy slightly. While using

all features from Dataset 1 with XGBoost, the accuracy obtained is 98.583 percent which is slightly increased. Considering only 15 out of 79 features with a difference of 0.225 percent, the performance of the model using IG and XGBoost for Dataset 1 is quite high. Figure 2(b) shows the performance ML models using TreeSHAP. The graph shows similar pattern with IG but with a slightly increased accuracy. XGBoost outperformed other classifiers at each number of features used. The highest accuracy obtained by XGBoost is 98.561 percent using 15 and 20 features, showing an increase of 0.203 percent compared to IG. The accuracy shows only an insignificant drop of 0.022 percent when all 79 features were used.

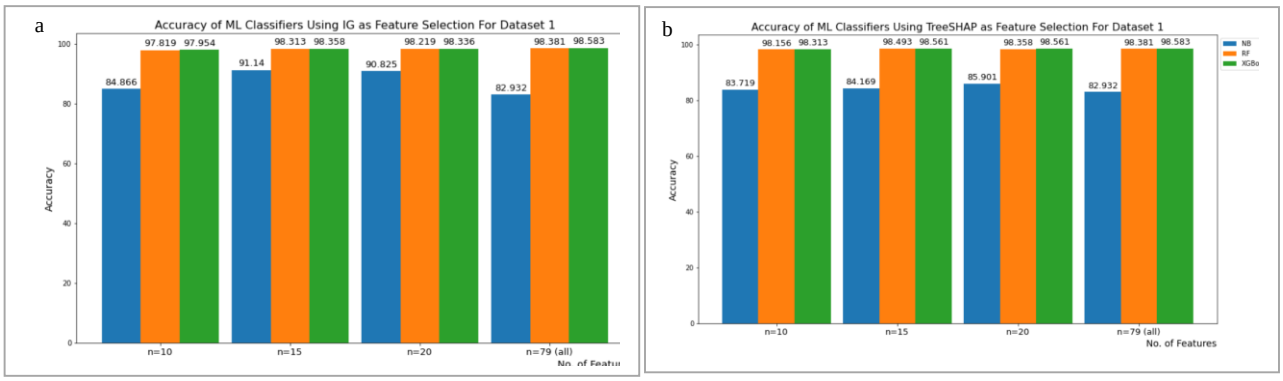

Fig. 2 - (a) Performance of classifies using IG as feature selection for dataset 1; (b) performance of classifies using TreeSHAP as feature selection for dataset 1

Figure 3(a) shows the performance accuracy of IG for Dataset 2. In contrast to the first dataset, RF has the best performance for every increase in features. The highest accuracy of 86. 64 percent is obtained when the number of features is 20. While using all features, XGBoost achieved the highest accuracy of 91.049 percent. The difference between the accuracy of using 20 features and all features is 4.409 percent. The low performance accuracy is observable for Naive Bayes every time the features are increased. Figure 3(b) shows the performance of TreeSHAP for Dataset 2. TreeSHAP's feature selection has better performance when compared to IG. The highest accuracy was obtained using 20 features selected by TreeSHAP with an accuracy of 90.81 percent. Using all features, the accuracy shows an insignificant increase of 91.049 percent using XGBoost.

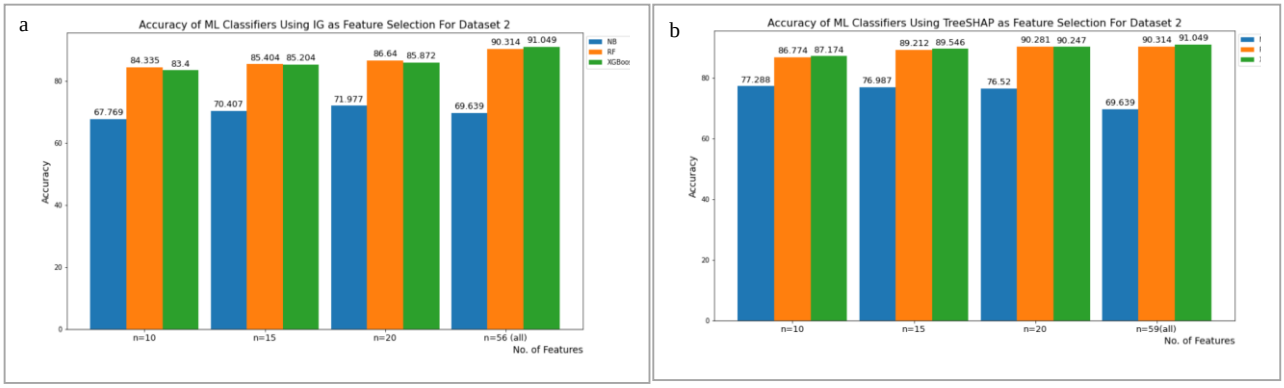

Fig. 3 - (a) Performance of classifies using IG as feature selection for dataset 2; (b) performance of classifies using TreeSHAP as feature selection for dataset 2

4.7 Comparison with Other Approach

This section compares the techniques used in this study with other existing phishing website detection. The comparison is assessed based on performance accuracy, category of the features used, and number of features as shown in Table 12.

Table 12 - Comparison of results with previous studies

Dataset	Approach	Category of Feature	Feature Selection Technique	Number of Selected Features	ML Classifier	Accuracy
D1	UÇAR [17]	URL	-	79	CNN	93.67
	Bu & Kim [25]	URL	GA	-	Deep Learning	96.85
	Proposed Method *	**URL**	**TreeSHAP**	**15**	**XGBoost**	**98.561**
	Kassim [24]	URL	SAE+PCA	20	LightGBM	99.6
	Gupta et al. [11]	URL	Manually Using Spearman Correlation, K-Best, and RF	9	RF	99.57
D2	**Proposed Method ***	**URL**	**TreeSHAP**	**20**	RF	**90.281**
	Hannousse & Yahiouche [16]	URL	-	56	RF	91
	Hannousse & Yahiouche [16]	Hybrid	Chi-Square	77	RF	94.09
	Moedjahedy et al [26]	Hybrid	Combines correlation and RFE	10	RF	97.6

For the first dataset, this study achieved accuracy of 98.56 percent using 15 features selected by TreeSHAP. As a result, this technique outperformed UÇAR [17] and Bu & Kim [25] who obtained 93.67 percent and 96.85 percent accuracy. Despite this, my method cannot surpass Kasim [24] and Gupta [11]. Gupta et al. [11]obtained 99.57 percent accuracy with only 9 features, while Kasim [24] obtained 99.6 percent with 20 features. Unlike the technique used in this study, which used predefined features from the dataset, Kasim [24] and Gupta et al. [11] extracted their own set of features.

Based on 20 features selected by TreeSHAP, the accuracy for the second dataset is 90.28 percent. In this study, accuracy dropped 0.719 percent from work by Hannousse & Yahiouche [16]. Accordingly, the result is comparable with Hannousse & Yahiouche [16] considering only 20 URL-based features were used as opposed to Hannousse & Yahiouche [16] which used 56. Moedjahedy et al. [26] obtained the highest accuracy of 97.6 percent with 10 hybrid features. It is significant to note that Hannousse & Yahiouche [16] and Moedjahedy et al. [26] obtained a high degree of accuracy compared to this work, but their work is highly dependent on the content of the website and third-party services. Hannousse & Yahiouche [16] performed feature selection techniques using Chi-Square. They used all the hybrid dataset (URLs, Content, Third Party) and were able to increase accuracy to 94.09 with 77 features. While this study only used features based on URL to detect phishing websites.

5. Conclusion and Future Works

This study aimed to identify the feature selection model that can be used to detect phishing websites by using URL-based features using the ML technique. The datasets with 79 and 56 features were selected to identify which feature selection performed better in selecting the best feature subset. This study proposes a feature selection technique using TreeSHAP to select the most significant features that improve the effectiveness of phishing detection using ML techniques. Three popular classifiers have been used to evaluate the performance of the models, which are Naïve Bayes, Random Forest, and XGBoost. Random Forest is a very popular classifier that shows great performance in improving accuracy. While XGBoost has started gain the popularity as the performance is on the par with Random Forest. As for the Naïve Bayes, it shows a good performance with Dataset 1 but with Dataset 2. This because some performance of classifiers is dependent on the dataset being used. A comparison with the prior works shows that this technique is able to achieve comparable performance. In the future, this study aims to improve the effectiveness of phishing website detection using the latest dataset public dataset to evaluate and compare the performance and characteristic of the newest phishing techniques. Secondly, use combinations of future selection techniques in order to select lesser features and significant attributes that improve the effectiveness of the detection.

Acknowledgement

This research was supported by Universiti Tun Hussein Onn Malaysia (UTHM), Book Hack Enterprise through SEPADAN RE-SIP (vot M071) and Politeknik Sultan Mizan Zainal Abidin (PSMZA).

References

[1] Almseidin, M., Abu Zuraiq, A. M., Al-kasassbeh, M., & Alnidami, N. (2019). Phishing detection based on machine learning and feature selection methods. *International Journal of Interactive Mobile Technologies*, *13*(12), 71-183. https://doi.org/10.3991/ijim.v13i12.11411

[2] Das Guptta, S., Shahriar, K. T., Alqahtani, H., Alsalman, D., & Sarker, I. H. (2022). Modeling Hybrid Feature-Based Phishing Websites Detection Using Machine Learning Techniques. *Annals of Data Science*. https://doi.org/10.1007/s40745-022-00379-8

[3] Tharani, J. S., & Arachchilage, N. A. G. (2020). Understanding phishers' strategies of mimicking uniform resource locators to leverage phishing attacks: A machine learning approach. *Security and Privacy*, *3*(5), 1-14. https://doi.org/10.1002/spy2.120

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

[5] Goud, N. S., & Mathur, A. (2021). Feature Engineering Framework to detect Phishing Websites using URL Analysis. *International Journal of Advanced Computer Science and Applications*, *12*(7), 295-303. https://doi.org/10.14569/IJACSA.2021.0120733

[6] Zamir, A., Khan, H. U., Iqbal, T., Yousaf, N., Aslam, F., Anjum, A., & Hamdani, M. (2020). Phishing web site detection using diverse machine learning algorithms. *Electronic Library*, *38*(1), 65-80. https://doi.org/10.1108/EL-05-2019-0118

[7] Aljofey, A., Jiang, Q., Qu, Q., Huang, M., & Niyigena, J. P. (2020). An effective phishing detection model based on character level convolutional neural network from URL. *Electronics (Switzerland)*, *9*(9), 1-24. https://doi.org/10.3390/electronics9091514

[8] Rao, R. S., Vaishnavi, T., & Pais, A. R. (2020). CatchPhish: detection of phishing websites by inspecting URLs. *Journal of Ambient Intelligence and Humanized Computing*, *11*(2), 813-825. https://doi.org/10.1007/s12652-019-01311-4

[9] Das, A., Baki, S., El Aassal, A., Verma, R., & Dunbar, A. (2020). SoK: A Comprehensive Reexamination of Phishing Research from the Security Perspective. *IEEE Communications Surveys and Tutorials*, *22*(1), 671-708. https://doi.org/10.1109/COMST.2019.2957750

[10] Zabihimayvan, M., & Doran, D. (2019). Fuzzy rough set feature selection to enhance phishing attack detection. *ArXiv*.

[11] Gupta, B. B., Yadav, K., Razzak, I., Psannis, K., Castiglione, A., & Chang, X. (2021). A novel approach for phishing URLs detection using lexical based machine learning in a real-time environment. *Computer Communications*, *175*(November 2020), 47-57. https://doi.org/10.1016/j.comcom.2021.04.023

[12] Alkhalil, Z., Hewage, C., Nawaf, L., & Khan, I. (2021). Phishing Attacks: A Recent Comprehensive Study and a New Anatomy. *Frontiers in Computer Science*, *3*(March), 1-23. https://doi.org/10.3389/fcomp.2021.563060

[13] Odeh, A., Keshta, I., & Abdelfattah, E. (2021). Machine LearningTechniquesfor Detection of Website Phishing: A Review for Promises and Challenges. *2021 IEEE 11th Annual Computing and Communication Workshop and Conference, CCWC 2021*, 813-818. https://doi.org/10.1109/CCWC51732.2021.9375997

[14] Benavides, E., Fuertes, W., Sanchez, S., & Sanchez, M. (2020). Classification of Phishing Attack Solutions by Employing Deep Learning Techniques: A Systematic Literature Review. *Smart Innovation, Systems and Technologies*, *152*(Micrads), 51-64. https://doi.org/10.1007/978-981-13-9155-2_5

[15] Jalil, S., Usman, M., & Fong, A. (2022). Highly accurate phishing URL detection based on machine learning. *Journal of Ambient Intelligence and Humanized Computing*, *0123456789*. https://doi.org/10.1007/s12652-022-04426-3

[16] Hannousse, A., & Yahiouche, S. (2021). Towards benchmark datasets for machine learning based website phishing detection: An experimental study. *Engineering Applications of Artificial Intelligence*, *104*(December 2020), 104347. https://doi.org/10.1016/j.engappai.2021.104347

[17] UÇAR, E., UÇAR.Murat, & İNCETAŞ, M. O. (2019). A DEEP LEARNING APPROACH FOR DETECTION OF MALICIOUS URLS. *International Management Information Systems Conference "Connectedness and Cybersecurity," December 2019*, 11-20.

[18] Basit, A., Zafar, M., & Javed, A. R. (2020). *A Novel Ensemble Machine Learning Method to Detect Phishing Attack*. 4-8.

[19] Geyik, B., Erensoy, K., & Kocyigit, E. (2021). Detection of Phishing Websites from URLs by using Classification Techniques on WEKA. *Proceedings of the 6th International Conference on Inventive Computation Technologies, ICICT 2021*, 120-125. https://doi.org/10.1109/ICICT50816.2021.9358642

[20] Atari, M., & Al-Mousa, A. (2022). *A Machine-Learning Based Approach for Detecting Phishing URLs*. 82-88. https://doi.org/10.1109/idsta55301.2022.9923050

[21] Adi, S., Pristyanto, Y., & Sunyoto, A. (2019). The best features selection method and relevance variable for web phishing classification. *2019 International Conference on Information and Communications Technology, ICOIACT 2019*, 578-583. https://doi.org/10.1109/ICOIACT46704.2019.8938566

[22] Zaman, S., Uddin Deep, S. M., Kawsar, Z., Ashaduzzaman, M., & Pritom, A. I. (2019). Phishing Website Detection Using Effective Classifiers and Feature Selection Techniques. *ICIET 2019 - 2nd International Conference on Innovation in Engineering and Technology*, 23-24. https://doi.org/10.1109/ICIET48527.2019.9290554

[23] Gandotra, E., & Gupta, D. (2021). *An Efficient Approach for Phishing Detection using Machine Learning*. Springer Singapore. https://doi.org/10.1007/978-981-15-8711-5_12

[24] Kasim, Ö. (2021). Automatic detection of phishing pages with event-based request processing, deep-hybrid feature extraction and light gradient boosted machine model. *Telecommunication Systems*, *78*(1), 103-115. https://doi.org/10.1007/s11235-021-00799-6

[25] Bu, S. J., & Kim, H. J. (2022). Optimized URL Feature Selection Based on Genetic-Algorithm-Embedded Deep Learning for Phishing Website Detection. *Electronics (Switzerland)*, *11*(7). https://doi.org/10.3390/electronics11071090

[26] Moedjahedy, J., Setyanto, A., Alarfaj, F. K., & Alreshoodi, M. (2022). CCrFS: Combine Correlation Features Selection for Detecting Phishing Websites Using Machine Learning. *Future Internet*, *14*(8). https://doi.org/10.3390/fi14080229

[27] Lundberg, S. M., Erion, G. G., & Lee, S. (2019). *Consistent Individualized Feature Attribution for Tree Ensembles*. *2*. https://doi.org/10.48550/arXiv.1802.03888