INTERNATIONAL JOURNAL OF SOFTWARE ENGINEERING AND COMPUTER SYSTEMS
E-ISSN: 2180-0650
VOLUME #, ISSUE #, 2024, pp. 60 – 74
DOI: https://doi.org/10.15282/ijsecs.10.1.2024.6.0124

RESEARCH ARTICLE

Heterogeneous Ensemble Feature Selection: An Enhancement Approach to Machine Learning for Phishing Detection

Bamidele Musiliu Olukoya[1*], Gabriel Opeyemi Ogunleye[2], Patrick Olaniyi Olabisi[3], and Adesoye Sikiru Adegoke[4]

[1]Computing & Information Science, Bamidele Olumilua University of Education, Science, 360101 Ikere-Ekiti, Ekiti State, Nigeria
[2]Faculty of Science, Federal University of Oye-Ekiti, 360102 Ekiti-State, Nigeria.
[3]College of Engineering, Bells University of Technology, Ota, 112101 Ogun-State, Nigeria.
[4]College of Engineering Technology, Lagos State University of Science and Technology, 104101 Ikorodu, Lagos State, Nigeria

ABSTRACT - Presently, phishing attacks are recognized as a global pandemic, which is adversely affecting global security and causing setbacks to global economy. A successfully conducted phishing attack (cybercrime) results in devastating effects such as: bankruptcy for people and corporations, mostly leading to information and financial fatalities. In the pursuit of accurately providing solutions against phishing threats, machine learning techniques were found to be the right antidote in the detection processes. One of the most important sub-tasks in supervised ML models is feature selection as it helps to eliminate unnecessary features from the dataset without sacrificing data quality. Feature selection is a serious challenge in phishing detection and other classification tasks. The worth of the selected attributes/variables plays a key role in building powerful models and poor-quality data frustrates the process. This work explores the use of ensemble feature selection in data mining to select meaningful features. A novel feature selection technique for phishing detection is proposed, based on frequent, necessary, and correlated items. The innovative Heterogeneous Ensemble Feature Selection framework (HEFS) framework produced a new set of webpage features highly informative apart from the usual common features used for phishing detection. Two experiments were conducted in the process, and the results show that both the classical models and their ensemble versions performed amazingly well when evaluated on the baseline features compared to the component features. However, Boosted_NB recorded the highest accuracy of 0.974 (97.4%). The HEFS is highly recommended as an efficient feature selection method to detect correlated, frequent, and phishing-behaved features for machine learning-based detectors.

ARTICLE HISTORY
Received : 25 July 2023
Revised : 18 December 2023
Accepted : 27 September 2024
Published : 3 October 2024

KEYWORDS
Phishing detection
Cybersecurity
Machine learning
Feature selection
Ensemble

1.0 INTRODUCTION

The unique characteristic of technological development among others is the provision of simple means of accomplishing complex tasks, most especially in the area of communication. Technological development has created immeasurable opportunities for businesses around the world for people to market their products from their comfort zones, permitting unsolicited messages to be distributed [1,2]. These messages are not harmful but carry elements of distraction and annoyance as the recipients prefer them to seamlessly move into the spam folder. After the successful disbursement of spam on the network, the avenue was later hijacked by scammers to conduct illicit activities such as phishing. Phishing is a fraudulent practice of sending emails or other messages purporting to be from reputable firms in order to induce individuals and organisations to reveal sensitive information such as passwords, credit card numbers, manufacturing or industrial secrets, and so on [3]. Phishing attacks are usually perpetrated through emails or links sent to potential users to ferry them to a phony webpage as soon as the link is clicked [4]. These attacks are known to have grave consequences with the attacks resulting in severe economic damages and losses across the globe, which are shouldered by the potential internet users, businesses, and other institutions.

Presently, network security has become more porous with the release of new smart gadgets along with internet accessibility, which makes the job easier for phishers [5]. These gadgets gave criminals a great opportunity to effortlessly create phony emails and websites usually sent or hosted by genuine companies such as financial organizations and other relevant institutions dealing with sensitive data. The culprit generates and sends many phishing emails/links to numerous people on the network [6]. Once the recipients open the email or click on the link, the users are directed to a spoofed website where their sensitive information is harvested. The reports of Anti-phishing working groups and IBM revealed that phishing cases are skyrocketing on a daily routine. The cases of phishing attacks took another dimension during COVID-19 pandemic, as email platform was globally accepted to monitor and conduct major transactions. As the number of users drastically increased, the number of phishing attacks also rose as well [7]. Dutta [3], submitted that phishing attacks exponentially increased between January and March 2020 compared to the previous years. Likewise, the study in [8] affirmed that eighty-one percent of organizations around the world experienced phishing attacks in 2020, and the

***CORRESPONDING AUTHOR | B.M. Olukoya** | ✉ olukoya.musiliu@bouesti.edu.ng

number is envisaged to go higher based on the current situation and statistics. The situation of phishing attacks in digitalized society presently is seen to be highly provoking and critical. Phishing activities are shaking the entire world due to unstable behaviour in identifying phishing attacks. However, this requires reliable methods to effectively identify phishing attacks [9],[10].

However, the incessant occurrence of phishing activities brought about different efforts to curb the attacks and their cohorts through the development of anti-phishing agents. Anti-phishing agents are improvised techniques designed to intercept and block phony content or web pages. Generally, these anti-phishing techniques are classified into static and dynamic techniques [11] with the general-list methods being typical examples of static techniques. Although, the general-list techniques performed proficiently at the beginning of their deployment, but later became obsolete due to the new strategy adopted to conduct the attacks [7,12]. Although, most of the popular browsers and email service providers available built their filtering security tools on the general-lists approaches [13], the methods have several constraints that users find stressful. On the other hand, dynamic techniques are seen as effective methods in discerning whether an email or website is phishing or legitimate [14]. Currently, the machine learning (ML) approach is seen as the right antidote, as it uses its tools to consciously analyse the content of the email or address of the webpage to detect whether it is phishing or not [15].

ML is a kind of artificial intelligence that learns and makes judgments based on past data. The problem of phishing detection is a classification task and there is yet no universal solution produced to permanently address the emanated issues. It therefore has always attracted more methods that could bring better results. The unique characteristics of the ML technique such as scalability and adaptability enable it to address sophisticated email phishing attacks which were difficult for the traditional techniques [16]. However, the prior studies unveiled that the effectiveness of ML-based detectors is largely determined by the algorithm, and the quality of features chosen to represent the whole dataset [17]. Feature(s) are simply described as individual, measurable aspects or properties of an observed phenomenon. However, 'feature' and terms like 'variables,' 'attributes,' or 'predictors' are sometimes used interchangeably. The ML technique consists of several unique sub-steps, like data collection and representation, feature selection, mapping (training), and making a good classification. All these steps are crucial, but the most important is the feature selection phase. This is the phase where redundant, irrelevant, and superfluous features contained in the dataset are expunged. Other remarkable benefits of the feature selection steps to the ML model include: increase in model accuracy, with decrease in training time, complexity cost, interpretability, space, and the chance of overfitting [4]. Feature selection is a significant step, most especially for classification purposes, with the quality of the features selected playing a crucial role in building proficient models [18]. Most oftentimes, poor-quality data makes the processes of ML models more difficult, and high-dimensional features are not suitable for the models according to the existing studies.

Dealing with high-dimensional datasets is one of the main challenges of feature selection, due to the increasing amount of data generated on daily routine. It is ideal today because of the vast data to find ways to efficiently select the most relevant features to improve the performance of models [5,11,19]. Consequently, huge of these datasets contain redundant and superfluous features that negatively impact the performance of models and increase the risk of overfitting. Observably, a large number of the existing anti-phishing studies pay more attention to optimizing classification models without considering the quality of data to be inputted into the models [20, 21]. This shows that the majority of these models were exposed to poor data, which later hurt their performance. However, a reliable and effective feature analysis framework that can identify and select optimal features is required [22].

Several identified techniques for feature selection can be categorized as: filter, wrapper, and embedded methods [23]. The filter method adopts statistical and information theory to evaluate the predictive power of individual features without invoking any ML algorithm [24]. The wrapper method on the other hand applies a machine-learning model to evaluate the given set of features and the information obtained is used to guide the selection of the important features. The last category incorporates feature selection as part of the model training process and uses penalization to select a subset of features [18]. There are one or two limitations attached to each of the methods mentioned, and this gave birth to other feature selection strategies such as deep learning-based feature selection, ensemble feature selection, and feature selection with dimensionality reduction [19]. In machine learning, ensemble learning is a kind of a notable method of combining several models, also known as "base learners" or "weak learners," to enhance the accuracy of a predictive model. As the size of digitalized datasets continues to grow exponentially, feature selection methods that can effectively handle high-dimensional data have become increasingly important [25]. The filter method has its drawbacks, but when individual filters are combined, it tends to thrash the limitations. One can say that prior works that utilized filter measures such as information gain, ANOVA, and correlation on individual terms are incomplete. In many cases, a combination of methods may be used to obtain the best results. As a result of the importance of feature selection in building an efficient model, we introduce a new feature selection strategy for phishing attack detection in order to reduce the size of the subset of the selected features and increase the classifier's effectiveness without compromising its accuracy.

Therefore, as a contribution to tackling the ever-increasing growth and changing patterns of phishing attacks and cohorts, the researchers propose an innovative feature selection framework known as the Heterogeneous Ensemble Feature Selection framework. The HEFS framework applied three filter-based statistical techniques to select the optimal subset features. The HEFS baseline features were extended to an ensemble structure of three single classifiers: Naïve Bayes (NB), Support Vector Machine (SVM), and Logistic Regression (LR).

The following are the contributions made by this study;

1. developed an innovative feature selection framework that can produce a highly effective optimal feature subset of different datasets and eliminate correlated features.

2. identified the relevant webpage baseline features that largely contribute to future machine learning-based phishing detection techniques.
3. improved the detection accuracy of machine learning-based phishing detection solutions.

The remainder of this paper is organized as follows: Section 2 presents the related works. Section 3 describes the implementation methodology. Section 4 presents and discusses results and provides an analysis. Section 5 concludes the work and provides suggested future works.

2.0 RELATED WORKS

The performance of phishing detection strategy is often affected by the feature selection techniques applied. We discuss some of the recent studies on detecting phishing using machine learning strategies.

However, [26] proposed a method to minimize the features for training deep learning classifiers. The previous study was based on the full set, while the current study used 10 features selected by the information gain selector. They selected three deep learning models and were evaluated on the Rao and Pais dataset consisting of 3526 instances with 2119 and 1407 phishing and legitimate sites respectively. The result shows that information gain boosted the performance of LSTM and DNN when the features were reduced. The issue here is that Information gain as a feature predictor tends biased toward features with higher-ranking values. The study of [27] suggested a self-structuring neural network-based intelligent phishing detection system. The authors harvested 17 features from URLs, and source code to evaluate the system along with neural network. The backpropagation techniques were adopted for adjusting the weights of the network, giving the network the benefits of adapting to the changing features of phishing attacks. The proposed strategy resulted in a detection accuracy of 89.40%.

In addition, [28] proposed an optimization method for machine-learning techniques in detecting phishing attacks with the intention to improve the detection rate of the anti-phishing system for websites. The machine learning classifiers applied to classify legitimate and phishing websites include Naïve Bayes, ID3, K-NN, DT, and RF. The researchers applied Genetic Algorithms for the selection of subsets. The study affirms that there was a significant improvement when ML classifiers were integrated with GAs, ID3 classifier portrayed an accuracy value of up to 95% along with yet another generating Genetic Algorithms. A performance evaluation of ML models for webpage phishing attack detection was conducted in [29]. Three classical learning algorithms were modelled individually, and the models achieved an accuracy value of 93.39, 91.74 and 98.35% for K-NN, SVM and RF respectively. It was observed that RF obtained the highest accuracy among other models evaluated.

An investigation of the effectiveness of correlation-based heuristic feature evaluation on the performance of the phishing detection models was conducted in [15]. The method was tested on two popular supervised machine learning classifiers: SVM and Naïve Bayes. The dataset given to the models consists of both phishing and benign instances of 2541 and 2500 respectively. The experiment was conducted using a cross-validation strategy, and the paper submitted that the phishing classification for both NB and SVM achieved an astonishing performance of 0.04% False Positive and 89.96% accuracy respectively. A framework for website phishing attack detection based on a stacking ensemble model was proposed in [24]. Three filter-based and one rapper-based feature selection methods were used individually to select the important features, which include information gain, gain ratio, Relief-F, and recursive feature elimination (RFE). The algorithms used for the models are: NB, k-NN, SVM, RF Bagging, NN stacking1 [NN + RF + Bagging], and stacking2 [k-NN + RF + Bagging]. The researchers experimented with the phishing websites dataset available on the Kaggle having a record of 11,055 website instances and 32 features. After the comparison of the supervised algorithms and stacking model, the work submitted that stacking1 [NN + RF + Bagging] achieved an accuracy of 97.4%, which outperformed other models.

In [30], a method for protecting internet users from phony websites and any type of phishing attack was proposed by checking the conceptual and literal consistency of the ULR and the web content. Their technique obtained a 99.1% accuracy rate, which shows that it is sufficient for identifying various types of phishing attacks. The work of [20] presented a novel feature selection framework termed Hybrid Feature Selection for the selection of efficient indicators. The method was applied to the URL phishing variables, the baseline features were evaluated using six (6) machine learning classifiers. The experiment shows that only the Random Forest classifier performed remarkably compared to other struggling classifiers with an accuracy of 94.12%. However, the time taken by the model to compile is minimal compared to the optimized and hybridized models.

In addition, from the study of [31] a Convolutional Neural Network (CNN) model was proposed while a sequential pattern was applied to take account of the uniform resource locator information. The proposed model achieved an accuracy of 98.58, 95.46, and 95.22 %, respectively, based on the benchmark datasets used. A study on "Detection of Phishing Websites using an Efficient Feature-Based Machine Learning Framework" was conducted in [7]. The study was commissioned to protect email and internet users from all phishing vulnerabilities. They came up with a classification model inspired by the heuristic feature that is mined from the website domain, URL, web protocol, and source code to abolish the limitations of the existing phishing detection techniques. Their model used lists-based methods with heuristic, visual similarity feature extraction methods, and machine learning algorithms such as Logistic Regression, Decision Trees, K-Nearest Neighbours, and Random Forests, and compare the results to find the most efficient machine learning scheme**.** The model is trained and tested on 75:25 training and testing data ratio. The outcome revealed that the KNN algorithm has a performance accuracy of 90.7% in detecting phishing websites, while the effect of feature selection techniques was also acknowledged.

The authors in [32] proposed a combinational feature selection based on correlation coefficient and mutual information to evaluate the relationship that exists between various features. Following the study of [33], support vector machine (SVM) was merged with feature weighting and feature selection strategies. The top-ranked K = 500 variables were taken, and the chi-square was applied to select a considerable number of features. The selected features were chosen through the feature weighting process. Likewise, [34] presents an ensemble feature selection method utilizing the minimum redundancy and chi-square strategies. The features that are correlated to each other were detected and eliminated using the chi-square test and the features with low redundancy were also chosen.

3.0 METHODS AND MATERIAL

3.1 Proposed Ensemble Feature Selection Framework

The feature selection step is paramount in a machine learning-based task so as to enhance the predictive ability, reduce model training time, and at the same time improve the interpretability. The overview framework of the ensemble feature selection for this study is presented in Figure 1. In the feature selection phase, the two types of ensemble techniques generally identified are homogenous and heterogeneous. In the homogeneous approach, the dataset provided is clustered into different partitions and applies a single feature predictor on the portioned data, while the heterogeneous applies multiple feature predictors on a dataset. Ensemble strategies have gained relevance in feature selection as the strength of multiple feature selection algorithms can be combined to produce better outcomes [6,14,20].

The proposed Heterogeneous Ensemble Feature Selection framework is described as follows: Let T denote the whole phishing samples in the dataset $\{q_1, q_2, \dots q_n\}$, $C = \{r_{1,}\, r_{2,} \dots r_{n,}\}$ be the class target and filter predictors denoted FP_k, FP_t, FP_j (GR PCC, & CHI2) respectively. For dataset D, the FP_k metric measures the value of the individual raw variable in T. The FP_k (Gain Ratio) is applied to measure the values of the phishing variables. Using Equations (1) to (8), a set of values $\{\emptyset_{1,}k, \emptyset_{2,}k, \dots., \emptyset_{j,}k,\}$ are generated and ranked according to their importance. The FP_k is the advance of information gain. The above procedure is performed for the FP_t (PCC) to measure and rank the values of the phishing dataset using Eqs (9), (10) and (11).

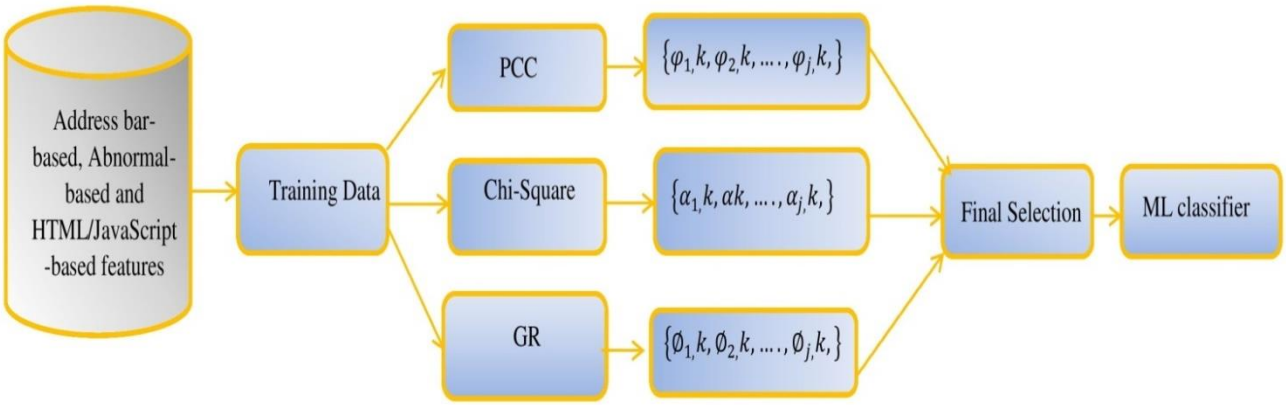

Figure 1. The Proposed HEFS Framework

The values $\{\alpha_{1,}k, \alpha k, \dots., \alpha_{j,}k,\}$ are generated using Equation (13). The three filter statistical techniques generate lists of values corresponding to the importance of the features $\{\emptyset_{n,}k, \varphi_{n,}k, \alpha_{j,}k,\}$ using FP_k, FP_t, FP_j respectively from the phishing dataset. A novel borda count aggregator is applied to the ranked $\{\emptyset_{n,}k, \varphi_{n,}k, \alpha_{j,}k,\}$ to obtain the reduced optimal subset features (baseline features). The Borda count is computed on the three sorted feature subsets from the FP_k, FP_t, FP_j respectively using Eq (12).

$$p_i = \frac{|c_i, T|}{|T|} \tag{1}$$

$$E(T) = -\sum_{i=1}^{m} p_i \log_2 p_i \tag{2}$$

where T is the phishing training set, p_i represents the probability that a sample in T belongs to a distinct class c_i, $E(T)$ represents the entropy of T, and m represents the total number of distinct classes in T.

Computing the values of information gain of features in the phishing data is to determine the expected reduction in entropy in each feature f_k, which involves the following steps:

a. Since the phishing data T is partitioned into subsets $T_{v,} = (1,2,3, \dots, p)$ based on the distinct values in f_k, the next action is to compute the entropy of each subset with respect to their class labels using Eqs. (3) and (4).

$$P_{Tv} = \frac{|c_{i,}\, T_v|}{|T_v|} \tag{3}$$

$$E(T_v) = -\sum_{i=1}^{m} P_{Tv} \log_2 P_{Tv} \tag{4}$$

b. Information Gain is computed using Eqs (5) and (6)

$$info_{fk}(T) = -\sum_{v=1}^{p} \frac{|T_v|}{|T|} E(T_v) \tag{5}$$

$$IG(f_k) = Ent(T) - info_{fk}(T) \tag{6}$$

where $IG(f_k)$ represents the information on each feature f_k, and p represents the number in which T is partitioned.

c. Finally, the Gain Ratio is computed by applying Eqs (7) and (8)

$$Splitinfo_{fk}(T) = -\sum_{v=1}^{n} \frac{|T_v|}{T} \log_2 \left(\frac{|T_v|}{|T|}\right) \tag{7}$$

$$FP_k = \frac{IG(T, f_k)}{SplitInfo_k(T)} \tag{8}$$

where $SplitInfo_{fk}(T)$ represent split information value generated by splitting the sample set T into p partitions corresponding to p distinct subsets on the feature f_k, and $G.R$ represents the Gain ratio which is the fraction of $IG(T, f_k)$ and $SplitInfo_{fk}(T)$.

$$FP_t\ (k,t) = \frac{cov(q,r)}{\sigma q \sigma r} \tag{9}$$

where q denotes the phishing independent variables, t is the phishing target class, $cov(k,t)$ present the covariance of q and r, $\sigma q, \sigma r$ is the standard deviation of q and r.

From Eq. 9, it is worthy of note that both the covariance and standard deviation needed to be computed. Hence, covariance and standard deviation were computed using Eqs. (10) and (11).

$$Cov(x\ , y) = \sum_{i=1}^{n} \frac{(q_i - \bar{q}_x)(r_i - \bar{r}_x)}{n-1} \tag{10}$$

$$\sigma_k = \sum_{h=1}^{n} \sqrt{\frac{(q-\bar{q})^2}{n-1}} \qquad and \quad \sigma_t = \sum_{h=1}^{n} \sqrt{\frac{(r-r)^2}{n-1}} \tag{11}$$

where $\bar{q}$ and r denotes the means of q and r.

Pearson's coefficient strength ranges from +1 to -1.

$$\begin{cases} +1\ q\ is\ positvely\ correlated\ to\ r \\ 0\ \ q\ is\ not\ correlated\ to\ r\ at\ all \\ -1\ q\ is\ negatively\ correlated\ to\ r \end{cases}$$

Hence, the ranking of the filter measure gives a coefficient, as features having a high correlation value are considered redundant features, and selected features are those having the minimum redundancy between consecutive features.

$$FP_j{}^2 = \sum_{i=1}^{m} \sum_{j=1}^{k} \left(\frac{A_{i,j} - \left(\frac{R_i * c_j}{N}\right)^2}{\frac{R_i * c_j}{N}} \right) \tag{12}$$

where m is the attributes magnitude in the phishing dataset; ***k* is the size of classes in the dataset**; N is the total size of samples in the dataset; R_i the size of patterns in the i^{th} attribute, C_j the size of patterns in the J^{th} class, and A_{ij} the size of patterns in the i^{th} internal and the J^{th} class.

The filter measures FP_k, FP_t, FP_j generates $\{\emptyset_{n,}k, \varphi_{n,}k, \alpha_{j,}k,\}$ from phishing samples in the dataset $\{q_1, q_2, \ldots q_n\}$, $C = \{r_{1,}\, r_{2,} \ldots r_{n,}\}$ respectively, Borda count applied Eq. 13 on the $\{\emptyset_{n,}k, \varphi_{n,}k, \alpha_{j,}k,\}$ to compute the final feature subsets.

$$b_i = \sum_{v]} N_f - P_v \tag{13}$$

Algorithm: Ensemble Feature Selection

Input:

FSM = $\{FSM_i | i = 1,2, \ldots . . q\}$; // where q = {Gain Ratio, Chi-Square, PCC}

Train D = $\{x_i y_i | i = 1,2, \ldots . . t\}$; // t denotes the number of phishing samples of train data

$F = \{f_1, f_2, \ldots f_{k=48}\}$; //k denotes the number of phishing features

Output: SubFeatures (SubFs)

Begin

for t = 1, 2, …, q do

Step 1: RankFs$_i$ = Feature selector (TrainD, FSM_i, k)

Step 2: Compute borda point = $b_i = \sum_{v]} N_f - P_v$

end for

Step 3: RankFs$_i$ based on Borda count in decreasing order

Step 4: Select final features based on a threshold value (SubFs)

End

3.2 Classification Algorithm

For the phishing webpage detection system, three machine-learning algorithms were selected based on their performances in classification problems, which are Logistic regression, Naïve, and Support Machine Vector. They are briefly discussed here.

a. Logistic Regression (LR):

Logistic regression is often used during classification tasks, due to the algorithm's ability to determine the mapping function between independent and specific dependent outcomes. It is one of the machine learning algorithms considered a baseline method for natural language processing [10]. Logistic regression is mostly used for solving binary problems, and can also be extended to solve multi-class problems. It is frequently applied to real-life applications such as spam filtering, ailment prediction (**health care**), **meteorology,** and sentiment tasks [1]. The logistic regression model relies on $logit$ transformation, written $logit(p)$, where p is the proportion of email dataset, with phishing and legitimate characteristics. To return the transformation categorical variable 0 (phishing) and 1 (legitimate) by $logit(p)$, we use Eq. (14):

$$\text{logit}(p) = \ln\left(\frac{p}{1-p}\right) \tag{14}$$

The value of $\text{logit}(p) = \beta$ is obtained as a linear combination of explanatory features, the actual value of the probability for the phishing variable is computed using Eq. (15):

$$p = \frac{e^{\beta}}{1+e^{\beta}} \tag{15}$$

Finally, to minimize the quadratic error function for logistic regression, Gradient descent is used as the optimization algorithm to leverage the cost value.

b. Naïve Bayes(NB)

Naïve Bayes classifiers are a collection of classification shallow learning schemes that rely on Bayes' Theorem with the assumption that there is no dependence existing among the predictors. Generally, the Bayes algorithm used conditional density $\boldsymbol{p(R|Q_1, \ldots, Q_p)}$ as pillar for target variable class $\boldsymbol{R = 1, .. K}$, given explanatory variables $\boldsymbol{Q_1}$ through $\boldsymbol{Q_p}$. Bayes theorem is computed using Eq. (16):

$$p(R|Q_1, \ldots, Q_p) = \frac{p(R)p(Q_1, \ldots, Q_p|R)}{p(Q_1, \ldots, Q_p)} \tag{16}$$

But the interest focuses on the numerator since the denominator is a constant $Q_j = q_j$ and the value of $R = r$ must be given. Although, $p(r)$ is not known, this represents the actual proportion of class r and plays the role of the prior probability. Given an outcome $\{q_1, q_2, \ldots, q_n\}$, the Bayes algorithm is the mode of (16):

$$argmaxt_r p(r)p(Q_1, \ldots, Q_p|R = r)$$

The conditional distribution for R now becomes;

$$p(R|Q_1, \ldots, Q_p) = \frac{1}{K(Q_1, \ldots, Q_p)} p(R) \prod_{i=1}^{p} p(Q_i|R),$$

where K is the normalizing constant parameter based on Q_1^p. The natural algorithm is the mode, given by:

$$\underset{r}{argmax}\, p(r) \prod_{j=1}^{p} p(q_j|r) \tag{17}$$

c. Support Vector Machine (SVM):

The support vector machine algorithm is a special class of universal network of feed-forward, used in real-life applications such as pattern classification. SVM is a linear classifier, that relies on structural risk minimization statistical learning theory, and other special features [16]. Hence, SVM is unique among other machine learning-based classifiers for its ability to provide a good generalization performance without incorporating problem-domain. The dilemmas found in multi-layer networks and single-layer neural network is settled through SVM or kernel machines. The main idea in building SVM classifier is the use of the inner-product kernel between a support vector and the input vector space. Let $\boldsymbol{q}$ denote certain input vector space and by $\boldsymbol{g(q) = \{gj(Q), j = 1, 2, \dots, x_1\}}$ denoted the non-linear transformation from the input space $\boldsymbol{q}$ to the feature space $\boldsymbol{f}$, dimension space $\boldsymbol{x_1}$. Given a linear transformation, SVM defines the hyperplane as a decision using equation 18

$$w_0 . q^T + b = 0 \tag{18}$$

where w denotes the weights vector, x_i denotes an instance from the phishing training dataset, and b represents the bias factor.

The distance from the vector q to the hyperplane, if the desired distance is denoted by v, then:

$$w_0 . q^T + b = r\|w\| \tag{19}$$

The value of the margin between the support vectors and the hyperplane is obtained by maximizing the distance:

$$\rho = \frac{2}{\|w\|} \tag{20}$$

Classification errors are inevitable when determining hyperplane given training phishing dataset, SVM introduces a function called slack variables in Eq. (21) to minimize the cost function:

$$L(w, \xi) = \frac{\|2\|^2}{2} + C . \sum_{i=1}^{N} \xi_{i,} \tag{21}$$

3.3 Boosting Ensemble

Shallow algorithms can be weak. To be weak simply means they may only do slightly better than random guessing at predicting specific target classes. The development of boosting techniques came onboard to improve the weak classifiers by iteratively optimizing each model on the initial dataset they trained. The iterative optimization uses (implicitly) an exponential loss function and a sequence of data-driven weights that increase the cost of misclassifications, thereby making successive iterates of the classifier more sensitive. The iterates form an ensemble of rules generated from a shallow classifier so that ensemble voting by a weighted sum over the ensemble usually gives better predictive accuracy.

Let's assume the phishing web page dataset $(q_1, r_1), \dots, (q_n, r_n)$, in which $q_1 \in E^p$ and $r_1 = 0,\ 1\ or\ 1, -1$. The number of iterations to improve in a given weak classifier $h_0(x)$ is given as integer K. At each iteration, a distribution in which to evaluate the misclassification error of h_t is required. The misclassification error is given as:

$$\varepsilon_t = PD_t(h_t(Q_i) \neq R_i) = \sum_{i=h_t(q_i)\neq r_i} D_t(i) \tag{22}$$

The probability under D_t that h_t misclassifies an q_i is set to be:

$$\alpha_t = \frac{1}{2} log \frac{1-\varepsilon_t}{\varepsilon_t} \tag{23}$$

Eq. (24) update D_t to D_{t+1} by:

$$D_{t+1}(i) = \frac{D_t(i) e^{-\propto_t r_i h_t(q_i)}}{C_t} \tag{24}$$

C_t is a normalization factor to ensure D_{t+1} is a probability vector.

The updated weighted vote classifier is computed using Eq. (25):

$$h_{t+1}(x) = -\left(\sum_{s=0}^{t+1} \alpha_t h_s^*(x)\right) \tag{25}$$

where, h_t is the boosted version of the initial classifier h_0.

3.4 Dataset

The dataset used for this experiment was made publicly available by [20]. The dataset was released for researchers and can be downloaded from the Kaggle machine learning community at Phishing Dataset for Machine Learning | Kaggle. The dataset is enormous and balanced, containing 5000 phishing and 5000 legitimate webpages based on URLs from PhishTank2, OpenPhish3, Alexa4, and Crawl5 archives respectively to be able to absorb the required features. The collection of the dataset was automated by using special tools, and related resources such as CSS, Javascript, and images were included aside from the HTML documents just to make them properly rendered in the browser. Consequently, the dataset was cleaned by discarding defective webpages in both legitimate and phishing, while duplicated instances were also expunged.

The dataset is suitable for the underlined study since it consists of new features that are proven to be relevant for identifying phishing attacks. The input and categorical features of the dataset are numerical. The CSV format was loaded into the Python environment where different experimental analyses were conducted. The phishing detection framework for this study is depicted in Figure 2.

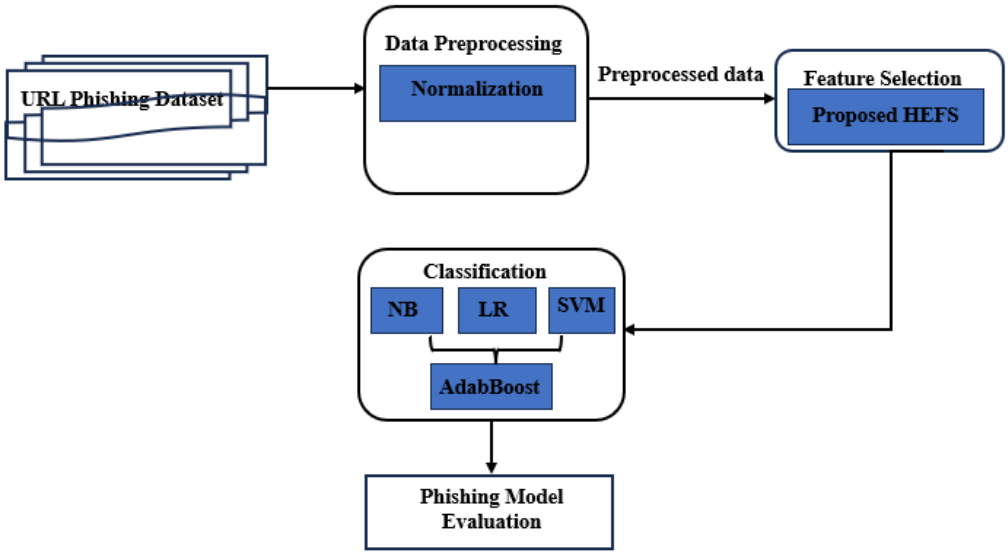

Figure 2. The Proposed Framework for Phishing Detection

3.5 Preprocessing

The researchers conducted exploratory data analysis with the goal to view the characteristics of the dataset. It was revealed from the analysis that there was no missing value as shown in Table 1 and Figure 3, but the values are skewed. As part of the preprocessing technique, the skewed values are scaled to achieve enhanced predictive ability by the min-max technique as in Eq. (26). The applied method scaled the variant values between 0 and 1. After data was rescaled, the proposed heterogenous ensemble feature selection was applied as shown in Figure 1. Both independent features that do not correlate to the target class and correlated independent variables were discarded to improve the predictive ability, reduce training time, and improve interpretability. For the models to have adequate familiarity with the set and to avoid model overfitting, the dataset was divided into two with a ratio of 80:20 for training and evaluation respectively.

$$V^i = \frac{V - V_{\min}}{V_{\max} - V_{\min}} \tag{26}$$

where, V is the new value to be converted, $V_{\min}$ is the minimum value, and $V_{\max}$ is the maximum value in the dataset.

Table 1. Phishing Characteristics

S/N	Characteristics	Values
1	Missing values	NIL
2	Independent variables	Numeric
3	Target class variables	Categorical
4	No of records (samples)	10000
5	No of features	48

3.6 Performance Evaluation Metrics

The performance of the phishing detection models is measured through evaluation metrics, namely: accuracy, precision, recall, specificity, and F1-Score. Each of these metrics is computed given the TP, TN, FP, and FN counts. To obtain the values of the performance metrics, the mathematical formulae given in Eqs (27) to (31) can be used.

$$\textbf{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{27}$$

$$\textbf{Specificity} = \frac{TN}{TN + FP} \tag{28}$$

$$\textbf{Precision} = \frac{TP}{TP + FP} \tag{29}$$

$$\textbf{Recall} = \frac{TP}{TP + FN} \tag{30}$$

$$\textbf{F1} - \textbf{Score (FS)} : = \frac{2 \times Prec \ + Recall}{Prec + Recall} \tag{31}$$

```
<bound method NDFrame.head of          id  NumDots  SubdomainLevel  PathLevel  UrlLength  NumDash  \
0         1        3               1          5         72        0
1         2        3               1          3        144        0
2         3        3               1          2         58        0
3         4        3               1          6         79        1
4         5        3               0          4         46        0
...     ...      ...             ...        ...        ...      ...
9995   9996        3               1          1         50        0
9996   9997        2               1          4         59        1
9997   9998        2               1          4         57        0
9998   9999        3               1          1         49        0
9999  10000        3               1          2         52        3

      NumDashInHostname  AtSymbol  TildeSymbol  NumUnderscore  ...  \
0                     0         0            0              0  ...
1                     0         0            0              2  ...
2                     0         0            0              0  ...
3                     0         0            0              0  ...
4                     0         0            0              0  ...
...                 ...       ...          ...            ...  ...
9995                  0         0            0              0  ...
9996                  0         0            0              0  ...
9997                  0         0            0              0  ...
9998                  0         0            0              0  ...
9999                  0         0            0              0  ...
```

Figure 3. Exploratory Data Analysis Showing No Missing Values

4.0 RESULTS AND DISCUSSION

4.1 Experimental Results

The experiment process was conducted in a Python environment because of its superb support libraries for machine learning. This study aims to show the impact of ensemble feature selection and monitor its effect on classical models and their ensemble without tuning the classifier's parameters. It is ideal to state that the classifiers used the default parameters. However, the majority of the prior studies conducted in the domain explored the benefits of optimization tools to trigger the detection rate of these classifiers.

Experiments were conducted on an Intel(R) Core(TM) i5-7200U CPU @ 2.50GH 2.70 GHz Laptop, CPU, 16.0GB RAM, and Windows 10Pro 64-bit operating system.

Experiment 1: Performance of the Classical Models on Individual Statistical Techniques

The experiments under experiment 1 were conducted using the features selected by individual filter-based statistical techniques, that is, Chi-Square, Pearson Correlation Coefficient, and Gain-ratio. The three selected classification algorithms built on each outcome, results obtained are given in Tables 2-4 and Figure 4-6. This is to facilitate the impact of the proposed feature selection method and that of its components.

Table 2. Results of NB, SVM, and LR using Chi-Square Features

PARAMETERS	**Single Models**			**Boosted Ensembled Models**		
	NB	**SVM**	**LR**	**NB**	**SVM**	**LR**
TRUE POSITIVE	906	897	466	849	869	921
FALSE-NEGATIVE	82	91	522	139	119	67
FALSE POSITIVE	305	362	23	262	352	308
TRUE NEGATIVE	707	650	989	750	660	704
Accuracy (%)	**0.807**	**0.774**	**0.728**	**0.799**	**0.764**	**0.813**
Fp Rate (Specificity)	0.699	0.642	0.977	0.741	0.652	0.696
Precision	0.748	0.712	0.953	0.764	0.712	0.749
Recall	0.917	0.908	0.472	0.859	0.879	0.932
F-Score	0.824	0.798	0.631	0.808	0.787	0.831

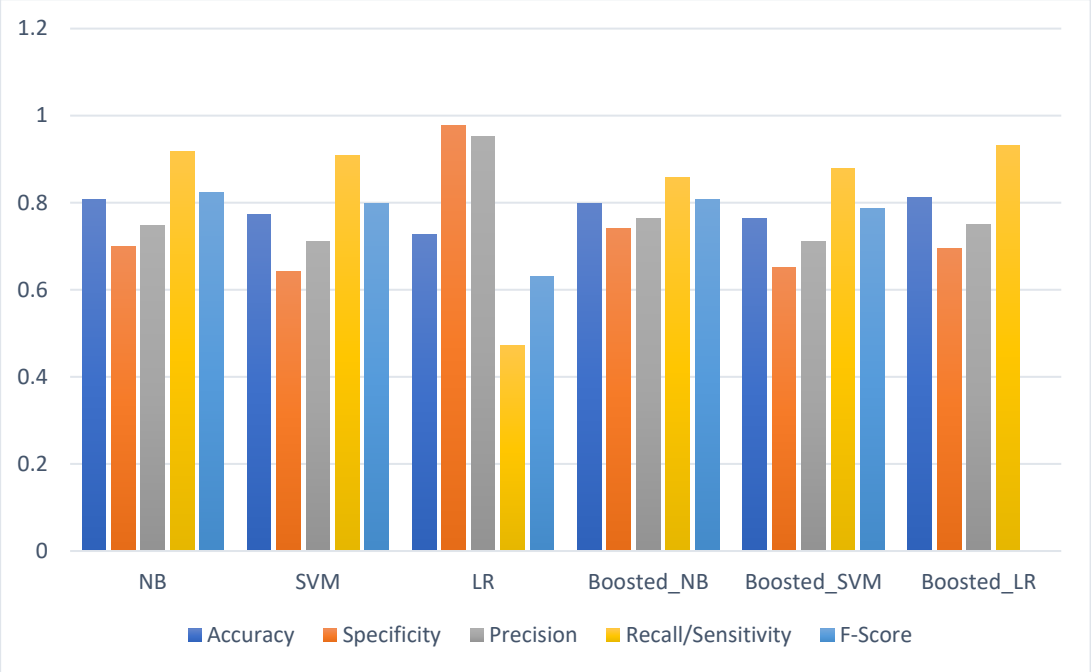

Figure 4. NB, SVM, LR and their Boosted Ensembled Using Chi-Square Features

Table 3. Results of NB, SVM and LR using Pearson Correlation Features

PARAMETERS	**Single Models**			**Boosted Ensembled**		
	NB	**SVM**	**LR**	**NB**	**SVM**	**LR**
TRUE POSITIVE	747	935	941	853	864	933
FALSE NEGATIVE	241	53	47	135	124	55
FALSE POSITIVE	81	377	315	155	351	318
TRUE NEGATIVE	931	635	697	857	661	694
Accuracy (%)	**0.839**	**0.785**	**0.819**	**0.855**	**0.763**	**0.813**
Fp Rate (Specificity)	0.920	0.627	0.689	0.847	0.653	0.686
Precision	0.902	0.713	0.749	0.846	0.711	0.745
Recall	0.756	0.946	0.952	0.863	0.875	0.944
F-Score	0.756	0.813	0.839	0.855	0.784	0.833

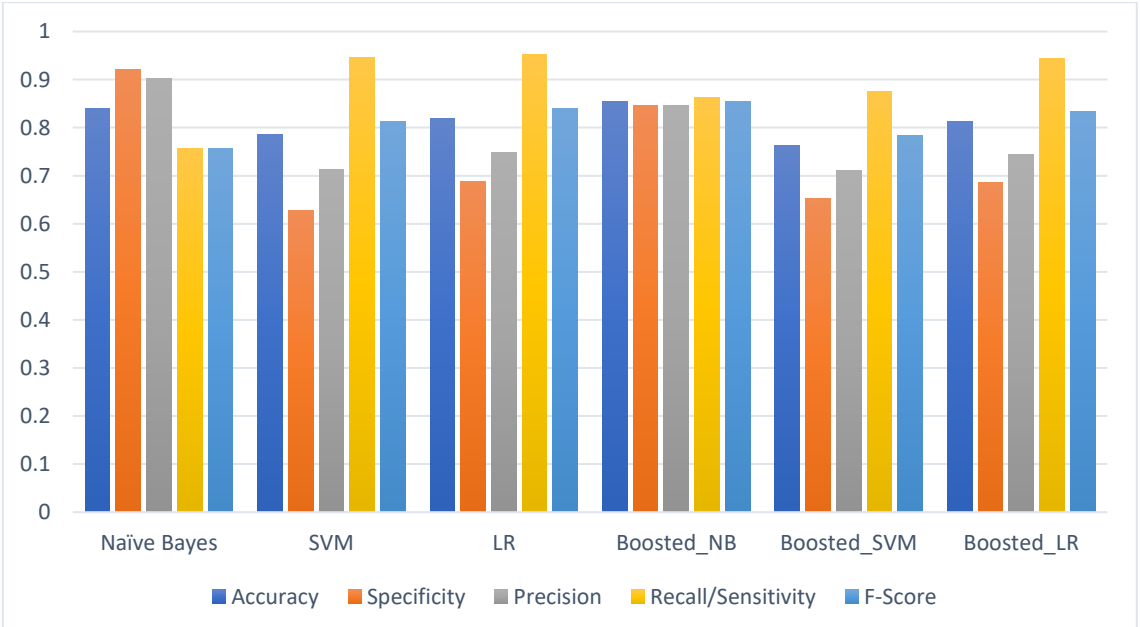

Figure 5. NB, SVM, LR, and their Boosted Ensembled Using Pearson Correlation Features

Table 4. Results of NB, SVM and LR using Gain Ratio Features

PARAMETERS	Single Models			Boosted Ensembled		
	NB	SVM	LR	NB	SVM	LR
TRUE POSITIVE	935	963	864	966	466	849
FALSE-NEGATIVE	53	25	124	22	522	139
FALSE POSITIVE	377	549	351	373	23	262
TRUE NEGATIVE	635	463	661	639	989	750
Accuracy (%)	**0.785**	**0.713**	**0.762**	**0.802**	**0.728**	**0.799**
Fp Rate (Specificity)	0.627	0.458	0.653	0.631	0.977	0.741
Precision	0.713	0.637	0.711	0.721	0.953	0.764
Recall	0.946	0.975	0.874	0.977	0.472	0.859
F-Score	0.813	0.770	0.784	0.830	0.631	0.808

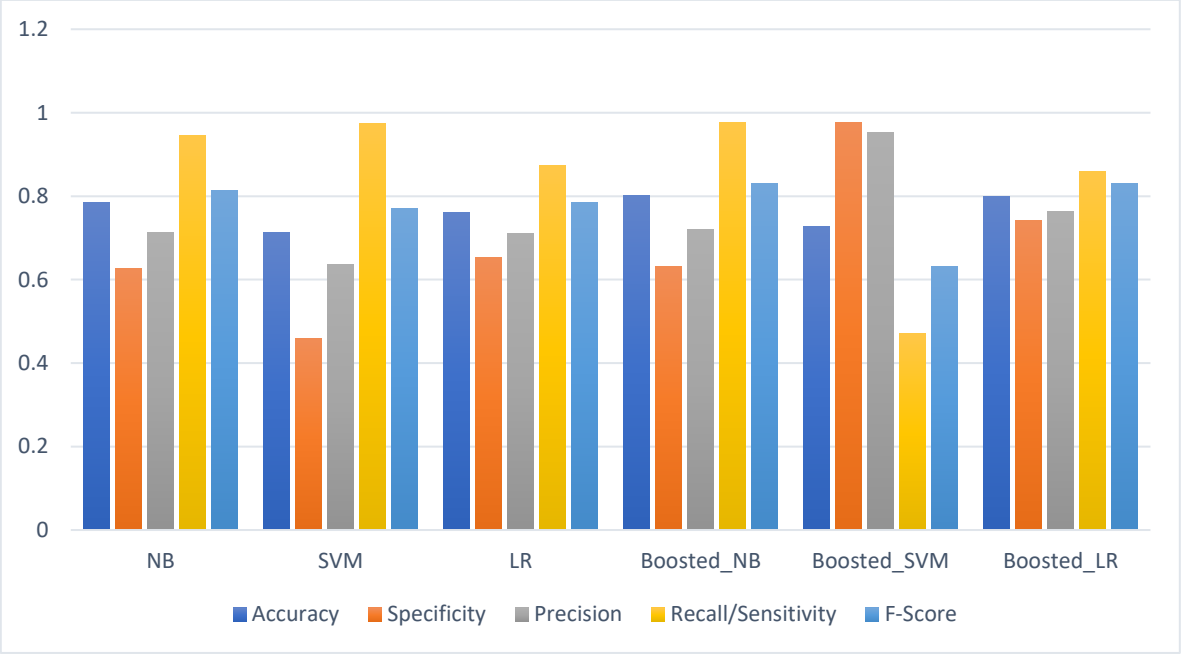

Figure 6. NB, SVM, LR and their Boosted Ensembled Using Gain Ratio Features

Experiment 2: Performance of the Classical Models on HEFS (Baseline) Features

The HEFS framework in Figure 1 was applied to the dataset to glean informative features from the vast features, it is expected that the feature selection should have better benchmark baseline features compared to the features selected by the framework components. The proposed HEFS framework selected 12 features from the cluster of Gain ratio, Pearson correlation, and Chi-square. The baseline features obtained from HEFS are presented in Table 5. The phishing detection model of SVM, LR, NB, and their boosted versions were built on the baseline features in the Table in the same way it was previously conducted on the components features, the performance results were presented in Table 6 and Figure 7.

Table 5. HEFS Baseline Features

FrequentDomainNameMismatch	PctExtNullSelfRedirectHyperlinks
PctExtResourceUrlsRT	NumDashInHostname
ExtMetaScriptLinkRT	PctNullSelfRedirectHyperlinks
SubmitInfoToEmail	FrequentDomainNameMismatch
NumSensitiveWords	PctExtHyperlinks
ExrFormAction	PctExtNullSelfRedirectHyperlinks

4.2 Discussion of Results

The researchers followed a funnel approach in conducting this study. Exploratory analysis was conducted on the dataset and it was revealed that there were no missing values, and the input features were in numeric format. The target class is categorical, and the skewed values were rescaled to improve the model's predictive ability. However, three

classical machine learning classifiers and their boosting versions were used and conducted in the Python environment. The selected algorithms were trained and tested based on 80:20 dataset split, the selected classifiers were trained on 80% and 20% for the testing. Two experiments were conducted, one part was conducted on the features obtained from the individual components that made up of the HEFS framework, and the second experiment was conducted on the baseline features, the outcomes of the models are compared. The results generated by the classical NB, SVM and LR, and their respective boosted ensemble models on the features obtained by individual statistical techniques are presented in Tables 2-4 and Figure 4-6.

Table 6. Results of NB, SVM, and LR using HEFS Features

PARAMETERS	**Single Models**			**Boosted Ensembled**		
	NB	**SVM**	**LR**	**NB**	**SVM**	**LR**
TRUE POSITIVE	908	912	901	968	913	918
FALSE-NEGATIVE	80	76	87	20	75	70
FALSE POSITIVE	72	64	130	32	45	59
TRUE NEGATIVE	940	948	882	980	967	953
Accuracy (%)	**0.924**	**0.930**	**0.892**	**0.974**	**0.940**	**0.936**
Fp Rate (Specificity)	0.929	0.937	0.872	0.968	0.956	0.942
Precision	0.927	0.934	0.874	0.968	0.953	0.939
Recall	0.919	0.923	0.912	0.979	0.924	0.929
F-Score	0.923	0.929	0.893	0.974	0.938	0.934

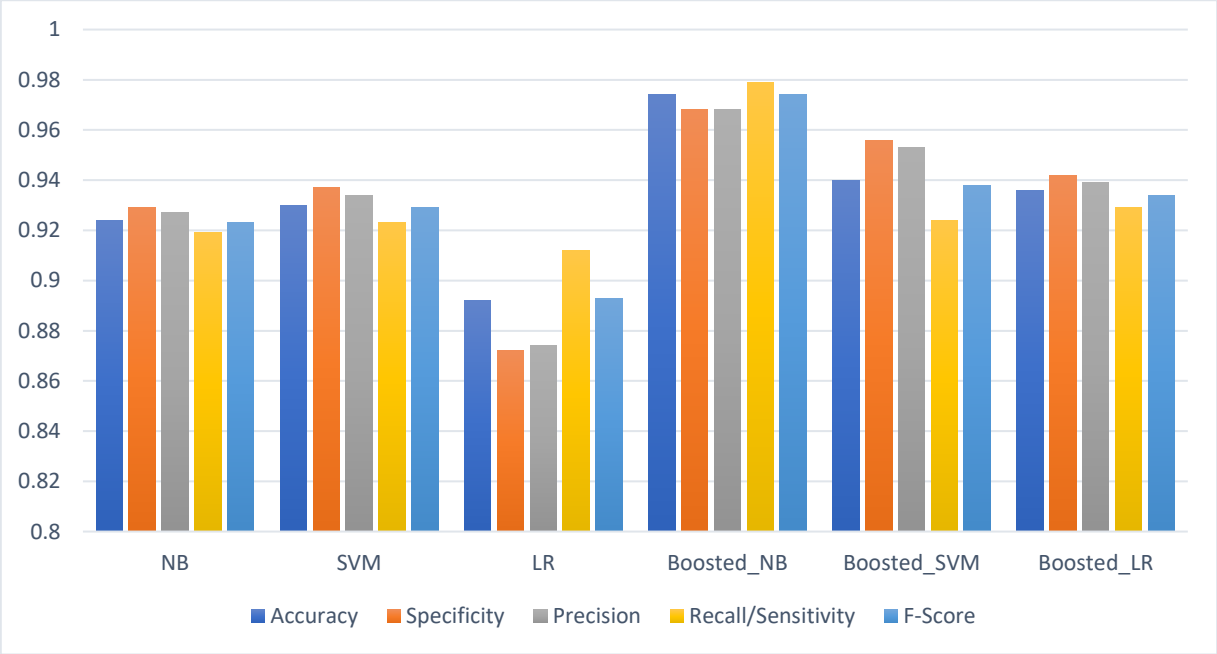

Figure 7: NB, SVM, LR and their Boosted Ensembled Using HEFS (Baseline) Features

The six models built in Table 2 using Chi-square features show that under the single classifiers, Naïve Bayes outperformed other models having 0.807(80.7%) with a recall of 0.917. Thereafter, after the models are boosted, it is discovered that Logistic regression outperformed the likes of Boosted_NB and Boosted_SVM with an accuracy of 0.813(81.3%). There are other metrics to review the performance of the models. Consequently, the same six models (SVM, NB, LR, and their various boosted forms) were built using Pearson Correlation features, and their performance results are presented in Table 3. The results revealed that both NB as a single learner and ensembled outperformed other models with an accuracy of 0.839(83.9%) and 0.855(85.5%) respectively. The performance of SVM was seen to drop from 0.785(78.5%) to 0.763(76.3%) when ensembled, likewise, LR slightly dropped. This is a pointer that variables selected by the PCC contained noise and irrelevant features. Naïve Bayes model had the highest precision of 0.902. The results presented in Table 4 are the performance of the models built on the best features selected by the Gain ratio. Analysing the results shows that NB recorded the highest accuracy of 0.785(78.5%), followed by LR at 0.762(72.6%). However, the performance of the NB was improved from 0.785(78.5%) to 0.802(80.2%). Thereafter, the performances of the two other models were drastically improved when they were ensembled.

The heart of this study relied on experiment 2, where the same classical models and their ensembled were trained and tested on the baseline features. The results harvested showed that there is astronomical improvement in the performance of the classical and their ensembled models under the baseline features compared to the previous results obtained under

the components features. The highest accuracy found from the single classical models is 0.930(93.0%) produced by SVM, followed by Naïve Bayes with an accuracy of 0.924(92.4%). The performances of these models were later improved when ensembled along with the baseline features, whereby the Boosted_NB achieved the highest accuracy of 0.974(97.4%0), Boosted_SVM 0.940(94.0%) and Boosted_LR 0.936(93.6%), respectively. In most of the previous studies, the performance of the classical SVM, NB, and LR does not usually reach 90%. Generally, the study provided empirical evidence that ensemble filter-based statistical techniques improve the performances of machine learning classifiers in phishing detection.

5.0 CONCLUSIONS

Each of the current feature selection methods has significant challenges. Thus, feature selection plays a significant role in machine learning's predictive efficacy. Feature selection method that can cope with large amounts of data and give the best features to the machine learning model to record high accuracy and operational efficiency is required for phishing detection systems. This study applied a novel Heterogeneous Ensemble Feature Selection, with respect to the theorem of no-free-lunch in searching for a possible method to curb phishing attacks. The innovative HEFS was implemented as a way to tackle the variants found in the individual feature predictors. The result of this method applied HEFS revealed and agreed with a few studies like [12,19,20], that features like NoHttps, NumDots, IpAddress, AtSymbol, QueryLength, MissingTitle, NumQueryComponents only have little contribution to the phishing detection techniques. This study has established the significant impact of the proposed feature selection method based on the results obtained from the evaluated models on the individual component's features and the baseline features. Phishing features are not static, the phisher often revisits their methods to release new sophisticated attacks to fool the existing agents. These features are no longer relevant and have become obsolete features in detecting phishing attacks. Thus, in this regard, anti-phishing researchers should discard these features and stop considering them as potential features for phishing detection. The results obtained from this experiment showed that the proposed feature selection framework proved to be more significant than the individual feature predictor approaches, whose computational time is high.

Consequently, this study has developed a reliable solution that can detect frequent features, features correlated to features, and target variables which can likewise eliminate features that are both redundant and unnecessary. This experiment can be enhanced further in the future, by investigating the HEFS framework using other classification classifiers or ensemble methods. Also, the proposed framework could be tested on more real-world datasets to achieve more confidence. This is to build optimistic confidence in the framework before planning to implement it in real-world systems that deal with phishing detection and other networks that use email services.

ACKNOWLEDGEMENTS

No grants were received from funding bodies in the public, private or not-for-profit sectors for this research except those deployed by the authors.

The authors would like to use this medium to thank everyone who contributed to this work most especially unidentified reviewers for their comments which helped raise the quality of this article.

AUTHORS CONTRIBUTION

B.M. Olukoya (Conceptualisation; Visualisation; Data acquisition; Writing - original draft)

G.O. Ogunleye (Validation; Data curation; Supervision)

P.O. Olabisi (Methodology; Investigation; Writing - review & editing)

A.S. Adegoke (Formal analysis; Software; Resources)

CONFLICT OF INTEREST

The authors declare no conflicts of interest whatsoever.

REFERENCES

[1] N. Bacanin *et al.*, 'Application of Natural Language Processing and Machine Learning Boosted with Swarm Intelligence for Spam Email Filtering', *Mathematics*, vol. 10, no. 22, Nov. 2022, doi: 10.3390/math10224173.

[2] S. A. Khan, W. Khan, and A. Hussain, 'Phishing Attacks and Websites Classification Using Machine Learning and Multiple Datasets (A Comparative Analysis)', *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 12465 LNAI, pp. 301–313, 2020, doi: 10.1007/978-3-030-60796-8_26.

[3] A. K. Dutta, 'Detecting phishing websites using machine learning technique', *PLoS One*, vol. 16, no. 10 October, Oct. 2021, doi: 10.1371/journal.pone.0258361.

[4] G. Hnini, J. Riffi, M. A. Mahraz, A. Yahyaouy, and H. Tairi, 'MMPC-RF: A deep multimodal feature-level fusion architecture for hybrid spam E-mail detection', *Appl. Sci.*, vol. 11, no. 24, Dec. 2021, doi: 10.3390/app112411968.

[5] A.M., Oyelakin, A. O. M, I. O, Mustapha, and I. K, Ajiboye, 'Analysis of Single and Ensemble Machine Learning

Classifiers for Phishing Attacks Detection', *Int. J. Softw. Eng. Comput. Syst.*, vol. 7, no. 2, pp. 44–49, 2021, doi: 10.15282/ijsecs.7.2.2021.5.0088.

[6] V. V. Ramalingam, P. Yadav, and P. Srivastava, 'Detection of Phishing Websites using an Efficient Feature-Based Machine Learning Framework', *Int. J. Eng. Adv. Technol.*, vol. 9, no. 3, pp. 2857–2862, Feb. 2020, doi: 10.35940/ijeat.C5909.029320.

[7] J. Zhou, H. Cui, X. Li, W. Yang, and X. Wu, 'A Novel Phishing Website Detection Model Based on LightGBM and Domain Name Features', *Symmetry (Basel).*, vol. 15, no. 1, Jan. 2023, doi: 10.3390/sym15010180.

[8] T. O. Omotehinwa and D. O. Oyewola, 'Hyperparameter Optimization of Ensemble Models for Spam Email Detection', *Appl. Sci.*, vol. 13, no. 3, Feb. 2023, doi: 10.3390/app13031971.

[9] F. Hossain, M. N. Uddin, and R. K. Halder, 'Analysis of optimized machine learning and deep learning techniques for spam detection', in *2021 IEEE International IOT, Electronics and Mechatronics Conference, IEMTRONICS 2021 - Proceedings*, Institute of Electrical and Electronics Engineers Inc., Apr. 2021. doi: 10.1109/IEMTRONICS52119.2021.9422508.

[10] M. Al-Sarem *et al.*, 'An optimized stacking ensemble model for phishing websites detection', *Electron.*, vol. 10, no. 11, Jun. 2021, doi: 10.3390/electronics10111285.

[11] O. Osanaiye, H. Cai, K. K. R. Choo, A. Dehghantanha, Z. Xu, and M. Dlodlo, 'Ensemble-based multi-filter feature selection method for DDoS detection in cloud computing', *Eurasip J. Wirel. Commun. Netw.*, vol. 2016, no. 1, Dec. 2016, doi: 10.1186/s13638-016-0623-3.

[12] C. M. Igwilo and V. T. Odumuyiwa, 'Comparative Analysis of Ensemble Learning and Non-Ensemble Machine Learning Algorithms for Phishing URL Detection', *FUOYE J. Eng. Technol.*, vol. 7, no. 3, pp. 305–312, 2022, doi: 10.46792/fuoyejet.v7i3.807.

[13] A. Taha, 'Intelligent ensemble learning approach for phishing website detection based on weighted soft voting', *Mathematics*, vol. 9, no. 21, Nov. 2021, doi: 10.3390/math9212799.

[14] R. P. Bellapu, R. Tirumala, and R. N. Kurukundu, 'Evaluation of homogeneous and heterogeneous distributed ensemble feature selection approaches for classification of rice plant diseases', *Proc. - 5th Int. Conf. Intell. Comput. Control Syst. ICICCS 2021*, no. Iciccs, pp. 1086–1094, 2021, doi: 10.1109/ICICCS51141.2021.9432081.

[15] A. A. Orunsolu, A. S. Sodiya, and A. T. Akinwale, 'A predictive model for phishing detection', *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 2, pp. 232–247, 2022, doi: 10.1016/j.jksuci.2019.12.005.

[16] R. Vinayakumar, K. P. Soman, Prabaharan Poornachandran, S. Akarsh, and M. Elhoseny, 'Deep learning framework for cyber threat situational awareness based on email and URL data analysis', in *Advanced Sciences and Technologies for Security Applications*, Springer, 2019, pp. 87–124. doi: 10.1007/978-3-030-16837-7_6.

[17] E. A. Amusan, O. T. Adedeji, O. Alade, F. A. Ajala, and K. O. Ibidapo, 'A Mobile Anti-Phishing System Using Linkguard Algorithm', *FUOYE J. Eng. Technol.*, vol. 6, no. 3, pp. 10–14, 2021, doi: 10.46792/fuoyejet.v6i3.666.

[18] H. M. Farghaly, A. A. Ali, and T. A. El-hafeez, 'Building an Effective and Accurate Associative Classifier Based on Support Vector Machine Building an Effective and Accurate Associative Classifier Based on Support Vector Machine', no. March, 2020.

[19] H. Mamdouh Farghaly and T. Abd El-Hafeez, 'A high-quality feature selection method based on frequent and correlated items for text classification', *Soft Comput.*, vol. 27, no. 16, pp. 11259–11274, 2023, doi: 10.1007/s00500-023-08587-x.

[20] K. L. Chiew, C. L. Tan, K. S. Wong, K. S. C. Yong, and W. K. Tiong, 'A new hybrid ensemble feature selection framework for machine learning-based phishing detection system', *Inf. Sci. (Ny).*, vol. 484, pp. 153–166, 2019, doi: 10.1016/j.ins.2019.01.064.

[21] J. Moedjahedy, A. Setyanto, F. K. Alarfaj, and M. Alreshoodi, 'CCrFS: Combine Correlation Features Selection for Detecting Phishing Websites Using Machine Learning', *Futur. Internet*, vol. 14, no. 8, Aug. 2022, doi: 10.3390/fi14080229.

[22] H. Abutair, A. Belghith, and S. AlAhmadi, 'CBR-PDS: a case-based reasoning phishing detection system', *J. Ambient Intell. Humaniz. Comput.*, vol. 10, no. 7, pp. 2593–2606, 2019, doi: 10.1007/s12652-018-0736-0.

[23] N. Noureldien and S. Mohmoud, 'The Efficiency of Aggregation Methods in Ensemble Filter Feature Selection Models', *Trans. Mach. Learn. Artif. Intell.*, vol. 9, no. 4, pp. 39–51, Aug. 2021, doi: 10.14738/tmlai.94.10101.

[24] A. Zamir *et al.*, 'Phishing web site detection using diverse machine learning algorithms', *Electron. Libr.*, vol. 38, no. 1, pp. 65–80, 2020, doi: 10.1108/EL-05-2019-0118.

[25] O. Osho, A. Oluyomi, S. Misra, R. Ahuja, R. Damasevicius, and R. Maskeliunas, *Comparative evaluation of techniques for detection of phishing URLs*, vol. 1051 CCIS. Springer International Publishing, 2019. doi: 10.1007/978-3-030-32475-9_28.

[26] M. Somesha, A. Roshan Pais, R. Srinivasa Rao, and V. Singh Rathour, 'Efficient deep learning techniques for the detection of phishing websites', 2046, doi: 10.1007/s12046-020-01392-4S.

[27] G. Mohamed, J. Visumathi, M. Mahdal, J. Anand, and M. Elangovan, 'An Effective and Secure Mechanism for Phishing Attacks Using a Machine Learning Approach', *Processes*, vol. 10, no. 7, Jul. 2022, doi: 10.3390/pr10071356.

[28] M. T. Suleman and S. M. Awan, 'Optimization of URL-Based Phishing Websites Detection through Genetic Algorithms', *Autom. Control Comput. Sci.*, vol. 53, no. 4, pp. 333–341, Jul. 2019, doi: 10.3103/S0146411619040102.

[29] R. S. Rao and A. R. Pais, 'An enhanced blacklist method to detect phishing websites', in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer Verlag, 2017, pp. 323–333. doi: 10.1007/978-3-319-72598-7_20.

[30] N. A. Azeez, S. Misra, I. A. Margaret, L. Fernandez-Sanz, and S. M. Abdulhamid, 'Adopting automated whitelist approach for detecting phishing attacks', *Comput. Secur.*, vol. 108, Sep. 2021, doi: 10.1016/j.cose.2021.102328.

[31] A. Aljofey, Q. Jiang, Q. Qu, M. Huang, and J. P. Niyigena, 'An effective phishing detection model based on character level convolutional neural network from URL', *Electron.*, vol. 9, no. 9, pp. 1–24, 2020, doi: 10.3390/electronics9091514.

[32] H. Zhou, X. Wang, and R. Zhu, 'Feature selection based on mutual information with correlation coefficient', *Appl. Intell.*, vol. 52, no. 5, pp. 5457–5474, 2022, doi: 10.1007/s10489-021-02524-x.

[33] U. I. Larasati, M. A. Muslim, R. Arifudin, and A. Alamsyah, 'Improve the Accuracy of Support Vector Machine Using Chi Square Statistic and Term Frequency Inverse Document Frequency on Movie Review Sentiment Analysis', *Sci. J. Informatics*, vol. 6, no. 1, pp. 138–149, 2019, doi: 10.15294/sji.v6i1.14244.

[34] A. Chaiban, D. Sovilj, H. Soliman, G. Salmon, and X. Lin, 'Investigating the Influence of Feature Sources for Malicious Website Detection', *Appl. Sci.*, vol. 12, no. 6, Mar. 2022, doi: 10.3390/app12062806.