# Accuracy of Phishing Websites Detection Algorithms by Using Three Ranking Techniques

**Badiea Abdulkarem Mohammed[1†] and Zeyad Ghaleb Al-Mekhlafi[2††],**
b.alshaibani@uoh.edu.sa ziadgh2003@hotmail.com
University of Ha'il, Computer Science and Engineering, Ha'il 81481, Saudi Arabia

**Abstract**
Between 2014 and 2019, the US lost more than 2.1 billion USD to phishing attacks, according to the FBI's Internet Crime Complaint Center, and COVID-19 scam complaints totaled more than 1,200. Phishing attacks reflect these awful effects. Phishing websites (PWs) detection appear in the literature. Previous methods included maintaining a centralized blacklist that is manually updated, but newly created pseudonyms cannot be detected. Several recent studies utilized supervised machine learning (SML) algorithms and schemes to manipulate the PWs detection problem. URL extraction-based algorithms and schemes. These studies demonstrate that some classification algorithms are more effective on different data sets. However, for the phishing site detection problem, no widely known classifier has been developed. This study is aimed at identifying the features and schemes of SML that work best in the face of PWs across all publicly available phishing data sets. The Scikit Learn library has eight widely used classification algorithms configured for assessment on the public phishing datasets. Eight was tested. Later, classification algorithms were used to measure accuracy on three different datasets for statistically significant differences, along with the Welch t-test. Assemblies and neural networks outclass classical algorithms in this study. On three publicly accessible phishing datasets, eight traditional SML algorithms were evaluated, and the results were calculated in terms of classification accuracy and classifier ranking as shown in tables 4 and 8. Eventually, on severely unbalanced datasets, classifiers that obtained higher than 99.0 percent classification accuracy. Finally, the results show that this could also be adapted and outperforms conventional techniques with good precision.

***Keywords:***
*Phishing websites; Supervised machine learning; Scikit Learn library; Deep learning; Classifiers.*

## 1. Introduction

Cybercrime has become a concern for a growing number 0f organizations and investigators in recent years. Phishing is a type of cybercrime that is regarded as the most pernicious thread in the web security fabric. Cybercriminals like deceptive phishing because it is significantly easier to persuade somebody to accept a harmful link in a potentially valid phishing email than to get past a computer's safeguards. Phishers can obtain background knowledge on the victim's personal and professional histories, preferences, and actions by using open information sources. Typically, via social media sites such as LinkedIn, Facebook, and Twitter. Phishing is a technique used by attackers to steal a user's credentials and data by impersonating legitimate websites and emails. This type of attack has become a source of concern due to the fact that it affects a large number of internet users and organizations. Phishing is a technique in which an attacker impersonates a legitimate website (LW) of a targeted organization and then distributes it to victims via fake or junk emails or URLs posted on social media, social networks, or any other medium of communication. This may cause victims to browse the URLs contained in those emails or posts, which will redirect them to the bogus website [1]. Strange punctuation or capital letters may also be used in phishing scams. However, not all phishing emails have spelling issues, so because a message appears to be well-written will not really indicate it's genuine. Before you believe the email is genuine, keep looking for other clues.

Despite the fact that there are numerous anti-phishing solutions available, phishers continue to lure victims. The Anti-Phishing Working Group (APWG) has seen a 69.5 percent increase in the number of unique PWs over the last five years, reporting approximately 785,920 unique PWs in 2018 [2]. According to the FBI, phishing activity cost the global economy 2.7 billion dollars in 2018 [3].

A thorough anti-phishing plan is required to combat the phishing problem. Employees must be trained on present phishing patterns to improve their chances of detecting and responding to phishing assaults. They may take steps to prevent malicious emails from being opened, as well as undertake malware clean-up and password overrides for those who have been hacked. Every corporation must have an email security policy which includes anti-phishing guidelines and defines appropriate email usage. Anti-phishing technology may built on artificial intelligence that can detect and prevent phishing information across whole organization's operational platforms and technologies.

Due to the following reasons, deceptive phishing attacks can still succeed

Manuscript received February 5, 2022
Manuscript revised February 20, 2022
https://doi.org/10.22937/IJCSNS.2022.22.2.34

- professional adversaries with financial motivations carry out the attacks,
- these adversaries take advantage of Internet users' ignorance and illiteracy regarding computers, and
- the adversaries are constantly refining their strategies for attracting new victims. Not only is phishing the preferred method of malicious internet users, but it is also the least effective method for the majority of internet users.

Automated methods for PW detection have been implemented to protect the public from criminal intrusions. Manually blacklisting URLs was the earliest method; later browsers used this blacklist to protect users from potential threats. These databases do not include newly launched PWs, and thus are ineffective in combating "zero hour" attacks, as centralized databases only added the majority of phishing links 12 hours after the phishing attacks occurred, [4]. SML algorithms have been used in recent research to detect website phishing. Numerous experiments have utilized classification methods, various phishing datasets, and predefined features [5-7]. SML refers to supervised machine learning, wherein the computers are taught using well-labelled training data and then anticipate output using that data. In comparison to traditional phishing website detection methods, intelligent phishing website proper technique based on supervised machine learning techniques are becoming prevalent. These systems are smarter and much more flexible to the Web environment. Furthermore, the most prevalent supervised machine learning approaches are validated and assessed in order to explore a most efficient intelligent machine learning technique to detect phishing sites.

Three open questions justified the current study. To begin, while classification accuracy is greater than 99.5 percent and a variety of algorithms are used (e.g., ensemble gradient boosting [8], statistical models or Logistic Regression [9], probabilistic algorithms or Bayesian Networks [10], and C4.5 classification trees [11]), there is no consensus on the optimal classification algorithm for classifying PWs on datasets with pr. Second, classification accuracy has been demonstrated using state-of-the-art methods on datasets with significant biases. One of the ensembles boost classifications, adaptive boosting, uses many classifiers to improve classification performance. AdaBoost is a method to create iterative ensembles. A Classification or Regression Trees is a machine learning prediction method. It illustrates how the value of a target attribute can be anticipated using other values. It's a decision tree with each fork separated into a regression model and a forecast again for target attribute at the end of each node. Stacking Ensemble Models combine prediction from numerous machine learning techniques on the very same dataset. A more balanced class composition results in superior results for the preferred class. Whether these results were discovered as a result of the dataset-dependent method's design or the research's use of state-of-the-art schemes remains an open question. To summarize, our search did not uncover sufficient comparative studies comparing the results of publicly available phishing datasets and their outlined characteristics in order to address the questions raised above. As such, the purpose of this study is to determine which classical classification algorithm is the most effective at identifying PWs across all publicly available datasets with pre-defined features.

Eight classic SML algorithms were compared on three publicly available phishing datasets, as detailed later in the methodology section of this study. These algorithms were discussed before on [20,44] and they are as follows:

1. AdaBoost (AB) [12],
2. Classification and Regression Tree (CART) [13,14,44],
3. Gradient Tree Boosting (GTB) [15,44],
4. Multilayer Perceptron (MLP) [16,44],
5. Naïve–Bayes (NB) [17,44],
6. Random Forest (RF) [18,44],
7. Support-Vector Machine (SVM) [19,44],
8. Stacking Ensemble Model (SEM) [20,21].

Each algorithm was evaluated on three datasets, ranked using three distinct ranking techniques, and compared using Welch's T-Test to determine the statistically significant difference.

The current study is organized as follows: The following section provides an overview of related work. Following that, the current study's methodology is described, followed by a presentation of the experimental findings. Finally, there is a conclusion.

## 2. Related Literature

The investigators of science have invested a lot of endeavours in detecting PWs. Approaches of heuristic-based blacklisting (more in Section 2.1) can be applied to this problem. SML approaches (Section 2.2) can also be employed in this instance. Deep learning can be used to intended destinations and accurate task results using unsupervised learning approaches. It cuts down on the time spent on feature extraction, which is one of the most time-consuming aspects of machine learning. Its design has now become responsive to new and capable of working on a variety of challenges as a result of ongoing training. Deep learning (Section 2.3) is as well [7]. In [44], most of these related researches are systematically presented as in Table 1.

Table 1: Schemes to the solution of the problem of detecting PWs that rely on classification

| Reference | Classifier | Dataset | | Accuracy |
|---|---|---|---|---|
| | | phish | legit | |
| [8] | Gradient Boosting | 100,000 | 1000 | 99.90% |
| [9] | Logistic Regression | 16,967 | 1,499,109 | 99.90% |
| [10] | Bayesian Network | 8,118 | 4,780 | 99.60% |
| [11] | C4.54 | 24,520 | 138,925 | 99.78% |
| [22] | Classic Perceptron | 990,000 | 10,000 | 99.49% |
| [23] | RF | 26,041 | 26,041 | 99.44% |
| [22] | Label Efficient Perceptron | 990,000 | 10,000 | 99.41% |
| [24] | Logistic Regression | 1,945 | 404 | 99.40% |
| [11] | SVM | 24,520 | 138,925 | 99.39% |
| [23] | Fast Decision Tree Learner (REP Tree) | 26,041 | 26,041 | 99.19% |
| [22] | Cost sensitive Perceptron | 990,000 | 10,000 | 99.18% |
| [23] | CART5 | 26,041 | 26,041 | 99.15% |
| [25] | RF | 2,141 | 1,918 | 99.09% |
| [23] | J486 | 26,041 | 26,041 | 99.03% |
| [26] | J48 | 11,271 | 13,274 | 99.01% |
| [26] | PART7 | 11,271 | 13,274 | 98.98% |
| [26] | RF | 11,271 | 13,274 | 98.88% |
| [27] | Gradient Boosting | 1,000 | 1,000 | 98,78% |
| [11] | NB | 24,520 | 138,925 | 98,72% |
| [11] | C4.5 | 356,215 | 2,953,700 | 98.70% |
| [20] | SEM | 5000 | 5000 | 98.58% |
| [23] | Alternating Decision-Tree | 26,041 | 26,041 | 98.48% |
| [27] | SVM (Linear) | 1,000 | 1,000 | 98,46% |
| [27] | CART | 1,000 | 1,000 | 98,42% |
| [28] | Adaptive Neuro-Fuzzy Inference System | 6,843 | 6,157 | 98.30% |
| [29] | RF | 1,541,000 | 759,000 | 98.26% |
| [25] | Logistic Regression | 2,141 | 1,918 | 98.25% |
| [23] | Random Tree | 26,041 | 26,041 | 98.18% |
| [27] | k-Nearest Neighbours | 1,000 | 1,000 | 98,05% |
| [29] | MLP | 1,541,000 | 759,000 | 97.97% |
| [26] | Logistic Regression | 11,271 | 13,274 | 97.70% |
| [25] | NB | 2,141 | 1,918 | 97.59% |
| [29] | k-Nearest Neighbours | 1,541,000 | 759,000 | 97.54% |
| [27] | SVM (Gaussian) | 1,000 | 1,000 | 97,42% |
| [29] | C5.08 | 1,541,000 | 759,000 | 97.4 |
| [20] | SEM | 30647 | 58000 | 97.39% |
| [30] | RF | 6,157 | 4,898 | 97.34% |
| [29] | C4.5 | 1,541,000 | 759,000 | 97.33% |
| [20] | SEM | 4898 | 6157 | 97.16% |
| [29] | SVM | 1,541,000 | 759,000 | 97.11% |
| [30] | MLP | 6,157 | 4,898 | 96.90% |
| [30] | Logistic Model Tree (LMT) | 6,157 | 4,898 | 96.87% |
| [30] | PART | 6,157 | 4,898 | 96.76% |
| [30] | ID39 | 6,157 | 4,898 | 96.49% |
| [31] | RF | 40,000 | 150,000 | 96.40% |
| [30] | Random Tree | 6,157 | 4,898 | 96.37% |
| [5] | RF | 5,000 | 5,000 | 96.17% |
| [25] | SVM | 2,141 | 1,918 | 96.16% |
| [29] | NB | 1,541,000 | 759,000 | 95.98% |
| [27] | NB | 1,000 | 1,000 | 95,97% |
| [30] | J48 | 6,157 | 4,898 | 95.87% |
| [32] | Logistic Regression | 20,500 | 15,000 | 95.50% |
| [30] | JRip10 | 6,157 | 4,898 | 95.01% |
| [33] | RF | 48,009 | 48,009 | 94.91% |
| [26] | SVM | 11,271 | 13,274 | 94.79% |
| [5] | C4.5 | 5,000 | 5,000 | 94.37% |
| [30] | Randomizable Filtered Classifier | 6,157 | 4,898 | 94.21% |
| [5] | JRip | 5,000 | 5,000 | 94.17% |
| [5] | PART | 5,000 | 5,000 | 94.13% |
| [34] | Extreme Learning Machines (ELM) | 2,784 | 3,121 | 94.04% |
| [30] | Stochastic Gradient Descent | 6,157 | 4,898 | 93.95% |

| | | | | |
|---|---|---|---|---|
| [30] | NB | 6,157 | 4,898 | 93.39% |
| [30] | Bayesian Network | 6,157 | 4,898 | 92.98% |
| [5] | SVM | 5,000 | 5,000 | 92.20% |
| [35] | Logistic Regression | 500,000 | 500,000 | 90.78% |
| [5] | NB | 5,000 | 5,000 | 84.10% |
| [26] | NB | 11,271 | 13,274 | 83.88% |

### 2.1 Review of Blacklisting and Heuristics-Based Research

This initiative to solve the problem by creating a centralized blacklist for PWs' URL (e.g., Google Safe Browsing API and PhishTank) was ineffective because it the process of detecting and reporting a malicious URL is time consuming and the PWs have a small lifetime (hours to days)" [36]. Thus, the scientific community implemented new URL detection methods for PWs. PWs. The blacklisting method entails determining which objects must be excluded. A blacklist is a list of suspect or hostile entities to whom system and network entry or operating rights should be prohibited.

The heuristic approach is better than blacklisting techniques where signatures of common attacks are compiled and used to blacklist new attacks [37]. Because blacklists are ineffective in detection of phishing sites due to its short lifespan, heuristics emerge as a preferred method at time 0. Heuristic methods can identify new URLs' threats and have greater generalization capabilities, however, they are not able to universally detect all threats [36].

### 2.2 Review of SML Based Research

During the past decade, most ML approaches for PWs detection used SML approaches on phishing datasets with predefined features. For phishing detection, machine learning can also be used in supervised machine learning. To train and validate the algorithms, separate datasets were created. The information was first separated into training and validation groups. The algorithms then were educated and assessed. Experts could use machine learning studio to experiment with decision trees and SVM and analyze the outcomes. This method of experimenting aided in the discovery of the optimal solution to the research topic. The resulting test results was then utilized to evaluate the training set [38,39]. Here, we summarize previous work done on this problem-solving topic over the last decade. Our review is composed of the publication year, authors, and number of phishing and LWs. accuracy is used to order results of these previous work.

This review provides the following observations:

- Two best schemes had an accuracy of 99.9%.
- Fifteen schemes achieved accuracies above 99.0%.
- RF (eight related studies), NB (seven related studies), SVM (seven related studies), C4.5 (seven related studies), Logistic Regression (six related studies) are the best-known algorithms among researchers.
- Five best schemes achieved 99.49% and more, were used with various kinds of classification: neural networks, Bayesians, ensembles, decision trees, and regression.
- In five approaches, the evaluation of classifier performance by precision is insufficient and does not say how this classification will work on more balanced datasets.

### 2.3 Review of Related Work of Deep Learning

Within the former few years, novel techniques to tackle PWs have been introduced by the scientific community. A Gated Recurrent Neural Network (GRU) that doesn't require any manual feature creation can successfully distinguish between phishing and LWs, correctly classifying all 240,000 URLs with 98.5 percent accuracy [31]. An experiment in which convolutional neural network (CNN) was utilized to design and extract features from general raw character strings (file paths, malicious URLs, etc.) for 19,067,879 randomly sampled websites URLs [38]. Authors of [39] conducted a comparative study, revealing that CNN and CNN Long Short-Term Memory (CNN-LSTM) deep learning networks achieve a 98.7% accuracy on 116,101 URL samples. Another Scheme that used binary classification is done using Deep Neural Networks (DNN) and Greedy Multilayer Deep Belief Network (DBN) can classify malicious URLs with 75% accuracy on 17,700 phishing URLs and 10,000 LWs [40].

## 3. Materials and Methods

Methodology of this work based solely in the methodology presented in [44]. In this section, research methodology was presented that include: experimental design, algorithms and basis for algorithm selection, datasets, and metrics (i.e. Classification Accuracy). Figure 1 shows the follow of the present research methodology.

### 3.1 Experimental Design

In the experiment, the classifiers were trained, then ranked, and then unified to form the classifier ranking. The Scikit Learn library includes eight popular classification algorithms that have been tuned for use with public phishing datasets. One of most widely used machine learning libraries is Scikit-learn. Many supervised or unsupervised deep learning are supported. It includes techniques for

grouping, analysis, and categorization, as well as other popular machine learning and data mining applications [43].

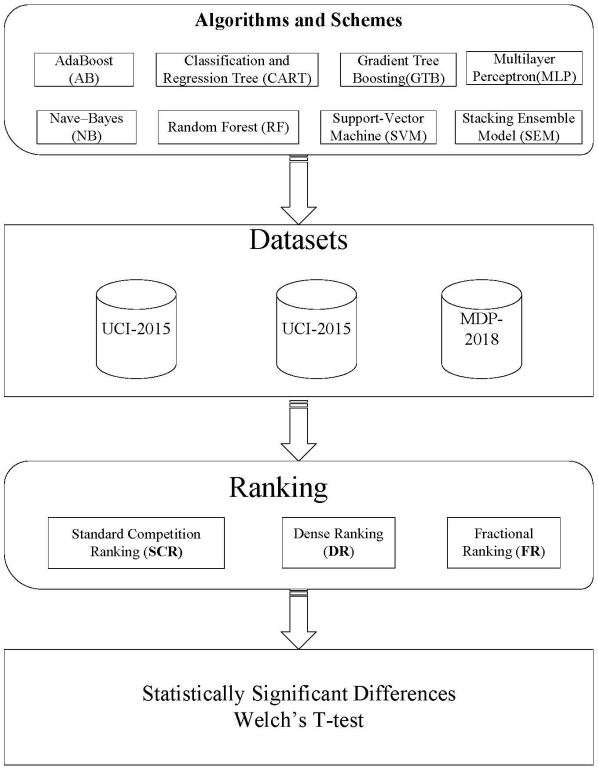


Figure 1 Methodological approach of the present study

In classifiers training part, all classifiers have been trained on three sets of data for their best accuracy of classification. The following steps were followed:

1. Use the Scikit Learn library to establish the classifier in Python for a certain data set [41].
2. Use the Cross Validation (CV) function of Scikit Learn to train and test the classifier with 30 stratified folds.
3. Draw curves of learning.
4. The following questions are to be answered to analysis learning curves and tuning hyper-parameters values
   - Does the algorithm learn or memorize data? The algorithm does not learn but stores the data when the training curve is flat at 100\%. To resolve this problem we take measures, for example reducing the weak learners' number in a set, reducing the tree depth, increasing the parameter for regularization, etc.
   - Are the algorithms likely to over-fit or under-fit (low distance, high variance) or do they just correctly learn? The algorithm is not adequate if the gap among the CV curves and the training is low; when the gap is large, the gap is over-fitting. We are working to resolve this problem by reducing high variance or bias such as (i) adding more examples, using small set of features, increasing the parameter of regularization, etc. in order to bring down high variance and (ii) using larger features set, adding polynomials characteristics, make more the neural network with more layers, decreasing the regularization parameter, etc. to make high bias reduction.

   If we choose to adjust the hyper-parameters values in order to avoid high variation or bias, we will start from step 2; otherwise, we will go to step 6.
5. Conduct the test of Wilk-Shapiro to test the distribution normality of the accuracies of the 30-fold CV testing classifier's classification. If not distributed normally, values normalization is required. The quality of the data supplied by the applicant in both the Profile Information Form and the resumes is cross-verified, and the legitimacy of the data is addressed during the CV verification process.
6. Keep the findings for another future actions.

This part is finished when all the classifiers are trained on every dataset and the distributed normality of the sets of classification accuracies is achieved.

In classifiers ranking part, Classifiers were ranked on the three datasets. The training data is made up of list of objects with a partial order defined between every list's contents. This ranking is usually established by assigning every item a number or numeric score or a binary judgement. The goal of the ranking system is to rank a permutation of things in fresh, unknown list in the same way as ratings inside the training examples are ranked [44,45]. The following steps are followed:

1. The Welch's T-test was employed to test the significant differences of classifications findings of each pair of classifiers. The classifications result should be normally distributed.
2. In descending order, classifiers were arranged according to their mean classification accuracy.
3. Each classifier was assigned three ranks using the ranking techniques. Classifiers with insignificant classifications results differences were assigned the same rank.

4. Determine the distribution of points for each classifier based on the received rank of every ranking technique, starting with the best ten points and decreasing to the lowest one. Points are calculated based on a mathematical formula (Equation 1).

$$P_i = N_m - R_i + 1, \quad (1)$$

where

$P_i$: ranking points,
$N_m$: number of ranking methods, and
$R_i$: the rank of method i.

5. Keep the findings for another future actions.

This part is completed after all classifiers have received ranking points from all datasets by all ranking methods.

Finally, ranking creation of unified classifier is the final step. This step aims to create a unified ranking that summarizes the performance of selected classifiers across all datasets. Combining rankings for each classifier produces a combined score for all datasets. This is the end of the experiment of the present study.

## 3.2 Algorithms and Schemes

Section 2.2 has five noteworthy SML algorithms and schemes: neural networks, decision trees, ensembles, regression, and Bayesian learning. Furthermore, the top three classifiers in terms of popularity are RF (8 papers), SVM (7 papers), and NB (7 papers). Accordingly, the used algorithms and schemes in the experiments of the present study were as follow:

1. The most Three algorithms in term of popularity.
2. Four best performing Scikit Learn library algorithms.
3. One state-of -art SEM.

Three public datasets with predefined features were used in the experiments of the present study. As the best of the authors' awareness, these are no other publicly available phishing datasets that contain predefined features. The UCI Machine Learning Repository is a library of data, subject ideas, and data sources that the machine learning group supports to test computational methods empirically. There are 6,157 phishing and 4,898 legal site entries in the UCI-2015 dataset. MDP-2018 is a symmetrical dataset that includes 5,000 samples of phishing and 5,000 samples of authentic websites. These sources yielded a total of 48 characteristics. These datasets are explained in Table 2.

Table 2: Datasets that used in the experiments

| Dataset | Creator name | Creation data | Number of PWs | Number of LWs |
|---|---|---|---|---|
| UCI-2015 | M. McCluskey (Univ. of Huddersfield) and Thabtah (Canadian Univ. of Dubai) | March 2015 | 6,157 | 4,898 |
| UCI-2016 | Abdelhamid (Auckland Institute of Studies) | November 2016 | 805 | 548 |
| MDP-2018 | C. L. Tan (Univ. Malaysia Sarawak) | March 2018 | 5,000 | 5,000 |

The following measures and methods were used to evaluate the results from the experiments of the present study: Classification accuracy is defined as the proportion of PWs and LWs that are classified correctly in comparison to all other websites. It can be mathematically represented as in Equation 2.

$$AC = \frac{TP+TN}{TP+TN+FP+FN} \quad (2)$$

where
AC: classification accuracy,
TP: true positive: number of successfully detected as PWs,
TN: true negative: number of successfully detected as LWs,
FP: false positive: number of incorrectly detected as PWs,
FN: false negative: number of incorrectly detected as LWs.
Classification accuracies were chosen as the classification quality quantification metric due to the following reasons:

1. Many other researchers evaluate results by examining classification accuracy. Therefore, their research results are comparable with results of the present study.
2. The distribution of used datasets in the present study was equal or near equal, therefore, majority and minority classes were not concerns of this study.
3. With the use of stratification option, cross-validation functions create test sets in which everyone have the same classes' distribution, or near distribution.
4. To separate the top classifiers, we employ ranking techniques. In these conditions, accuracy is an effective measure of bias.

The Welch's T-test is utilized in the experiments of the present study to assess whether any two classifiers' means of accuracy of classification are statistically different. The unpaired two-sample T-test is specified in [42]. Suppose two independent samples $X_{1,\cdots,}X_n$; and $Y_{1,\cdots,}Y_m$ have means $\mu_{x,\cdots,}\mu_y$; then, the test hypothesis is set to be: $H_0$: $\mu_x = \mu_y$ vs. $H_A$: $\mu_X \neq \mu_y$. The hypothesis is tested as follow:

$$T = \frac{\bar{X} - \bar{Y}}{\sqrt{\frac{S_x^2}{n} + \frac{S_y^2}{m}}} \quad (3)$$

where

$\bar{X}$ and $\bar{Y}$ are the two independent samples,
$n$ and $m$ are the samples sizes of $\bar{X}$ and $\bar{Y}$ respectively,
$S_x^2$ and $S_y^2$ are the samples variances.

In the present study, it is concluded that if $|T| > t_1 - \alpha/2, v$ with $(\alpha = 0.05)$, then we cannot accept the null hypothesis that emphasize the equal of the two means. Welch's T-test can run only on samples that normally distributed. Using the package of "scipy.stats", T-test was implemented in Python. SciPy Stats can create random numbers that are continuous or categorical. It also has a number of other routines for generating explanatory statistical information. One can work with continuous flow, randomized, and arbitrary data.

To check the normal distribution of the population, the Shapiro–Wilk test is used [43]. The test can be performed by the following formula that showed in Equation 4.

$$W = \frac{(\sum_{i=1}^{n} a_i x_i)^2}{\sum_{i=1}^{n} (x_i - \bar{x})^2} \quad (4)$$

where

$W$ is the Shapiro–Wilk value,
$a_i$ are coefficients of statistics order of size n sample obtained from normal distribution,
$x_i$ is an ascending order of sample values,
$\bar{x}$ is mean of sample $X$, and
$n$ is size of sample $X$.

If $W < W_\alpha$, then the zero hypothesis ($H_0$) that the sample is normally distributed is rejected. The Shapiro–Wilk test was performed with "scipy.stats" Python library.

In the present study, ranking techniques were employed as follow:

1. Standard Competition Ranking (SCR): In this technique, same ranking number is assigned to equal items and then left a gap in the numbers of ranking. For example, "1 2 2 4".
2. Dense Ranking (DR): In this technique, same ranking number is assigned to equal items and next ranking number is given for the next order. For example, "1 2 2 3". Each item's rank amount is equal to 1 plus the number of items listed above the ranking order that seem to be distinctive.
3. Fractional Ranking (FR): In this technique, all equally-rated items with identical numbers of ranking that is the means of their ordinal ranks. For example, "1 2.5 2.5 4". Items that contrast equally obtain the same ranking number, which would be the average of what they would get if they were ranked ordinally; correspondingly, the ranking number of 1 plus the quantity of parts placed from above plus half the quantity of parts equivalent to it.

Classification accuracy metric and balanced datasets were used in the present study. Construct validity of classification accuracy is high when balanced datasets are used. To assess classification accuracy, the cross-validation process with 30 stratified folds were employed, which reflect how objective measure model fits well and generalizes to new data. The Welch's T-test was performed to see if there was a statistically significant difference between the means of accuracy finding of classification that provided by each two classifiers in a specific dataset. This test ruled out the possibility of incorrectly ranking classifiers whose results were not statistically significant. To avoid ranking prejudice, three different ranking techniques were employed, each of which yielded different results. We utilized the source code available in "https://github.com/PauliusVaitkevicius/Exp001" [44].

## 4. Results and Discussion

The present section designates each method's results before comparing them with the related works with proper discussion.

### 4.1 Experimental Results in Training Stage

A classification algorithms were selected for each dataset. Open-source Python (version 3.7.1) and the library of Scikit Learn (version 0.20.1) implementations were used to implement all algorithms and selected functions [41]. 30-fold cross validation was performed to choose the best hyper-parameters for each algorithm on each dataset. Each classifier's Hyper-parameters are listed in Table 3. Different algorithm configurations are applied to datasets with varying designs and data quantities due to use of the hyper-parameter selection technique.

Table 3: Hyper-parameters for each algorithm

| Algorithm | UCI-2015 | UCI-2016 | MDP-2018 |
|---|---|---|---|
| AB | # of estimators: 200 | # of estimators: 50 Algorithm: SAMME; | # of estimators: 200 |
| CART | Min samples at leaf node: 2 Split evaluation criteria: entropy; Max tree depth: 9; | Min samples at leaf node: 2; Split evaluation criteria: entropy; Max tree depth: 9; | Min samples at leaf node: 2; Split evaluation criteria: entropy; Max tree depth: 5; |
| GTB | Learning rate: 1; Max estimator depth: 1; | Learning rate: 1; # of estimators: 50; Min samples at leaf node: 2 | Learning rate: 1; Max estimator depth: 1; |
| MLP | Number of max iterations: 3000; Hidden layers: 30; | of max iterations: 1000; Hidden layers: 30; | Number of max iterations: 1000; Hidden layers: 30; |
| NB | Multivariate Bernoulli; models; | Multivariate Bernoulli; models; | Multivariate Bernoulli; models; |
| RF | Split evaluation criteria: entropy; Max tree depth: 11; | Split evaluation criteria: entropy; Max tree depth: 8; | Split evaluation criteria: entropy; Max tree depth: 11; |
| SVM | Penalty parameter C: 1.0; Kernel: Linear | Penalty parameter C: 1.0; Kernel: Linear | Penalty parameter C: 1.0; Kernel: Linear |
| SEM | max depth: 142; Criterion: entropy; max features: 'auto'; | max depth: 142; Criterion: entropy; max features: 'auto'; | max depth: 142; Criterion: entropy; max features: 'auto'; |

The classifiers were trained and tested on all the datasets. Performance of classification was measured by measuring the legitimate and phishing links ratio in the dataset. Table 4 contains classification results. Initial results show that SEM did well on MDP-2018, UCI-2016, and UCI-2015 dataset respectively.

Table 4: Classification accuracy of different algorithms

| Algorithm | UCI-2015 | UCI-2016 | MDP-2018 |
|---|---|---|---|
| AB | 0.9352 | 0.8495 | 0.9728 |
| CART | 0.9363 | 0.893 | 0.9574 |
| GTB | 0.9381 | 0.9034 | 0.9742 |
| MLP | 0.9722 | 0.9028 | 0.9671 |
| NB | 0.9057 | 0.8225 | 0.9177 |
| RF | 0.9525 | 0.8916 | 0.9715 |
| SVM | 0.9271 | 0.8365 | 0.9422 |
| SEM | 0.98.58 | 0.9716 | 0.9876 |

Welch's T-test was performed on all classifications for each dataset to see if they differences were statistically significant. Three different techniques for sorting classifiers were used to arrange the classifiers on each dataset. Classifiers that are found with no significant differences, were given equal ranks. Next, each classifier was given points from the 10th to the 1st rank. Results of ranking over the UCI-2015, UCI-2016, and MDP-2018 dataset are shown in Tables 5-7 respectively.

Table 5: Rankings of classifiers on UCI-2015

| SCR rank | FR rank | DR rank | Algorithm | SCR points | FR points | DR points |
|---|---|---|---|---|---|---|
| 1 | 1 | 1 | ESM | 10 | 10 | 10 |
| 1 | 1 | 1 | MLP | 10 | 10 | 10 |
| 2 | 4.5 | 2 | RF | 9 | 6.5 | 9 |
| 2 | 4.5 | 2 | GTP | 9 | 6.5 | 9 |
| 2 | 4.5 | 2 | CART | 9 | 6.5 | 9 |
| 2 | 4.5 | 2 | AB | 9 | 6.5 | 9 |
| 8 | 8.5 | 3 | SVM | 3 | 2.5 | 8 |
| 10 | 10 | 4 | NB | 1 | 1 | 7 |

Table 6: Rankings of classifiers on UCI-2016

| SCR rank | FR rank | DR rank | Algorithm | SCR points | FR points | DR points |
|---|---|---|---|---|---|---|
| 1 | 2.5 | 1 | ESM | 10 | 8.5 | 10 |
| 1 | 2.5 | 1 | GTP | 10 | 8.5 | 10 |
| 1 | 2.5 | 1 | MLP | 10 | 8.5 | 10 |
| 1 | 2.5 | 1 | CART | 10 | 8.5 | 10 |
| 1 | 2.5 | 1 | RF | 10 | 8.5 | 10 |
| 5 | 7 | 2 | AB | 6 | 4 | 9 |
| 5 | 7 | 2 | NB | 6 | 4 | 9 |
| 10 | 10 | 3 | SVM | 1 | 1 | 8 |

Table 7: Rankings of classifiers on MDP-2018

| SCR rank | FR rank | DR rank | Algorithm | SCR points | FR points | DR points |
|---|---|---|---|---|---|---|
| 1 | 2 | 1 | ESM | 10 | 9 | 10 |
| 1 | 2 | 1 | GTP | 10 | 9 | 10 |
| 1 | 2 | 1 | AB | 10 | 9 | 10 |
| 1 | 2 | 1 | RF | 10 | 9 | 10 |
| 5 | 5.5 | 3 | MLP | 7 | 7 | 9 |
| 7 | 7.5 | 4 | CART | 6 | 5.5 | 8 |
| 9 | 9 | 5 | NB | 2 | 2 | 6 |
| 10 | 10 | 6 | SVM | 1 | 1 | 5 |

Finally, the combined dataset rankings are shown in Table 8, which calculates the various sets of algorithms' scores to see which classifiers ended up placing first. Utilizing the technique of Standard Competition Ranking, we found SEM and RF at the top. Using the Fractional Ranking method, SEM was also ranked at the top. Using the Dense Ranking technique, we get RF, SEM, and MLP, at the top. There is no algorithm that is number one in all three ranking techniques.

Table 8: Composed classifier rankings

| Algorithm | SCR points | FR points | DR points |
|---|---|---|---|
| ESM | 27 | 25.5 | 29 |
| MLP | 27 | 25.5 | 29 |
| GTP | 29 | 24 | 29 |
| RF | 29 | 24 | 29 |
| AB | 25 | 19.5 | 28 |
| CART | 25 | 20.5 | 27 |
| SVM | 16 | 13 | 25 |
| NB | 9 | 7 | 22 |

## 5. Conclusions

The purpose of this study is to answer the following question: Which classical classification algorithm is the best for detecting PWs on all publicly available datasets with predefined features? As a result, the following conclusions are drawn: To begin, neural networks, particularly SEM, RF, AB, and GTP are the most effective at detecting PWs. Second, regardless of dataset design, Bayesian and instance similarity-based classifiers (SVM and NB) perform poorly at detecting PWs. Third, the conclusions above are consistent with previous work, which stated that the best classification results are obtained when ensemble classification schemes, neural networks, and decision trees are used. Finally, classifiers that achieved greater than 99.0 percent classification accuracy on highly unbalanced datasets in the literature review, such as RF, SVM, MLP, and CART, did not achieve this level of accuracy in our experiments using balanced datasets.

## References

[1] B. B. Gupta, N. A. Arachchilage and K. E. Psannis, "Defending against phishing attacks: Taxonomy of methods, current issues and future directions," Telecommunication Systems, vol. 67, no. 2, pp. 247–267, 2018.

[2] APWG, "Phishing activity trends report," [Online]. http://www.antiphishing.org/APWG_Phishing-Activity_Report_Jul_05.pdf, 2005. [Accessed in 28 Jun 2021].

[3] S. S. Smith, "2017 Internet crime report," Federal Bureau of Investigation, Washington, DC. [Online]. https://www.ic3.gov/Media/PDF/AnnualReport/2017_IC3Report.pdf, 2018. [Accessed in 28 Jun 2021].

[4] A. K. Jain and B. B. Gupta, "A machine learning based approach for phishing detection using hyperlinks information," Journal of Ambient Intelligence and Humanized Computing vol. 10, pp. 2015–2028, 2019.

[5] K. L. Chiew, C. L. Tan, K. Wong, K. S. Yong and W. K. Tiong, "A new hybrid ensemble feature selection framework for machine learning-based phishing detection system,". Information Sciences, vol. 484, pp. 153–166, 2019.

[6] S. Marchal, K. Saari, N. Singh and N. Asokan, "Know your phish: Novel techniques for detecting phishing sites and their targets," in Proc. ICDCS, Nara, Japan , pp. 323–333, 2016.

[7] D. Sahoo, C. Liu and S. C. H. Hoi, "Malicious URL detection using machine learning: A survey," arXiv:cs.LG/1701.07179, vol. 1, no.1 pp. 1-37, 2019.

[8] S. Marchal, G. Armano, T. Gröndahl, K. Saari, N. Singh et al., "Off-the-hook: An efficient and usable client-side phishing prevention application," IEEE Transactions on Computers, vol. 66, pp. 1717–1733, 2017.

[9] C. Whittaker, B. Ryner, and M. Nazif, "Large-scale automatic classification of phishing pages," in Proc. (NDSS), San Diego, CA, pp. 1-14, 2010.

[10] G. Xiang, J. Hong, C. P. Rose and L. Cranor, "CANTINA+: A feature-rich machine learning framework for detecting phishing web sites," ACM Transactions on Information and System Security, vol. 14, no. 21, pp. 1-28, 2011.

[11] B. Cui, S. He, X. Yao and P. Shi, "Malicious URL detection with feature extraction based on machine learning," International Journal of High Performance Computing and Networking, vol. 12, pp. 166–178, 2018.

[12] R. Wang, "AdaBoost for feature selection, classification and its relation with SVM, a review," Physics Procedia, vol. 25, pp. 800–807, 2012.

[13] L. Breiman, J. Friedman, C. J. Stone and R. A. Olshen, "Classification and regression trees; CRC press," Boca Raton, Florida: CRC Press, 1984.

[14] W. Y. Loh, "Classification and regression trees," WIREs Data Mining and Knowledge Discovery, vol. 1, pp. 14–23, 2011.

[15] J. H. Friedman, "Stochastic gradient boosting," Computational Statistics and Data Analysis, vol. 38, pp. 367-378, 2002.

[16] B. Widrow and M. Lehr, "30 years of adaptive neural networks: perceptron, madaline, and backpropagation," Proceedings of the IEEE, vol. 78, pp.1415–1442, 1990.

[17] D. D. Lewis, "Naive (Bayes) at forty: The independence assumption in information retrieval," in Proc. ECML-98, Chemnitz, DE, pp. 4-15, 1998.

[18] A. Cutler, D. R. Cutler and J. R. Stevens, "Random forests," in Ensemble machine learning; Boston, MA: Springer, pp. 157–175, 2012.

[19] B. Schölkopf, A. J. Smola, F. Bach, "Learning with kernels: support vector machines, regularization, optimization, and beyond," London, England: MIT press, 2002.

[20] M. Al-Sarem, F. Saeed, Z. G. Al-Mekhlafi, B. A. Mohammed, T. Al-Hadhrami et al., "An optimized stacking ensemble model for phishing websites detection," Electronics, vol. 10, no. 11, pp. 1-18, 2021.

[21] Z. G. Al-Mekhlafi, B. A. Mohammed, M. Al-Sarem, F. Saeed, T. Al-Hadhrami et al. "Phishing websites detection by using optimized stacking ensemble model," Computer Systems Science and Engineering, Accepted on Jun 2021, pp.1–17, 2021. doi: 10.32604/csse.2021.020414.

[22] P. Zhao, S. C and Hoi, "Cost-sensitive online active learning with application to malicious URL detection," in Proc. KDD13, New York, NY, USA, pp. 919-927, 2013.

[23] D. R. Patil and J. B. Patil, "Malicious URLs Detection Using Decision Tree Classifiers and Majority Voting Technique," Cybernetics and Information Technologies, vol.18, no.1, pp.11–29, 2018.

[24] T.C. Chen, T. Stepan, S. Dick and J. Miller, "An anti-phishing system employing diffused information," ACM Transactions on Information and System Security. vol.16, no 4, pp. 1-31, 2014.

[25] A. K. Jain and B. B. Gupta, "Towards detection of phishing websites on client-side using machine learning based approach," Telecommunication Systems, vol. 68, no. 1, pp. 687–700, 2018.

[26] R. Verma and K. Dyer, "On the character of phishing URLs: Accurate and robust statistical learning classifiers," in Proc. CODASPY'15, New York, NY, USA, pp. 111-122, 2015.

[27] H. Shirazi, B. Bezawada, and I. Ray, "Know thy doma1n name: Unbiased phishing detection using domain name based features," in Proc. SACMAT '18, New York, NY, USA, pp. 69–75, 2018.

[28] M. Adebowale, K. Lwin, E. Sánchez and M. Hossain, "Intelligent web-phishing detection and protection scheme using integrated features of Images, frames and text," Expert Systems with Applications, vol. 115, pp. 300–313, 2019.

[29] F. Vanhoenshoven, G. Nápoles, R. Falcon, K. Vanhoof and M. Köppen, "Detecting malicious URLs using machine learning techniques," in Proc. SSCI, Athens, Greece, pp. 1-8, 2016.

[30] M. Karabatak and T. Mustafa, "Performance comparison of classifiers on reduced phishing website dataset," in Proc. ISDFS, Antalya, Turkey, pp. 1-5, 2018.

[31] J. Zhao, N. Wang, Q. Ma and Z. Cheng, "Classifying malicious URLs using gated recurrent neural networks," in Innovative Mobile and Internet Services in Ubiquitous Computing, Cham: Springer International Publishing, pp. 385–394, 2019.

[32] J. Ma, L. K. Saul, S. Savage and G. M. Voelker, "Beyond blacklists: Learning to detect malicious web sites from suspicious URLs," in Proc. KDD'09, New York, NY, USA, pp. 1245-1254, 2009.

[33] S. Marchal, J. François, R. State and T. Engel, "PhishStorm: Detecting phishing with streaming analytics," IEEE Transactions on Network and Service Management, vol. 11, pp.458-471, 2014.

[34] W. Zhang, Q. Jiang, L. Chen and C. Li, "Two-stage ELM for phishing web pages detection using hybrid features," World Wide Web, vol. 20, pp.797–813, 2017.

[35] K. Thomas, C. Grier, J. Ma, V Paxson and D. Song, "Design and evaluation of a real-time URL spam filtering service," in Proc. IEEE Symposium on Security and Privacy, Oakland, CA, USA, pp. 447–462, 2011.

[36] R. Verma and A. Das, "What's in a URL: Fast feature extraction and malicious URL detection," in Proc. IWSPA '17, New York, NY, USA, pp. 55-63, 2017.

[37] C. Seifert, I. Welch and P. Komisarczuk, "Identification of malicious web pages with static heuristics," in Proc. 2008 Australasian Telecommunication Networks and Applications Conference, Adelaide, SA, Australia, pp. 91-96, 2008.

[38] J. Saxe and K. Berlin, ``eXpose: A character-level convolutional neural network with embeddings for detecting malicious URLs, file paths and registry keys," 2017. [Online]. Available: https://arxiv.org/abs/1702.08568. [Accessed in 28 Jun 2021].

[39] A. Vazhayil, R. Vinayakumar and K. Soman, "Comparative study of the detection of malicious URLs using shallow and deep Networks," in Proc. ICCCNT, Bengaluru, India, pp. 1–6, 2018.

[40] S. Selvaganapathy, M. Nivaashini and H. Natarajan, "Deep belief network based detection and categorization of malicious URLs," Information Security Journal: A Global Perspective, vol. 27, no. 3, pp. 145–161, 2018.

[41] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, et al., "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.

[42] G. W. Snedecor and W. G. Cochran, “Statistical methods,” 8th Ed., vol. 54, Ames, IO, USA: Iowa State Univ. Press, pp. 71-82, 1989.
[43] S. S. Shapiro and M. B. Wilk, “An Analysis of variance test for normality (complete samples),” Biometrika, vol. 52, No. ¾, pp. 591-611, 1995.
[44] P. Vaitkevicius and V. Marcinkevicius, “Comparison of classification algorithms for detection of phishing websites.” Informatica, vol. 31, pp. 143–160, 2020.

**Badiea Abdulkarem Mohammed** received his BSc in Computer Science from Babylon University, Iraq in 2002, M.Tech in Computer Science from University of Hyderabad, India in 2007 and PhD from Universiti Sains Malaysia, Malaysia in 2018. He is currently an Assistant Professor in the College of Computer Science and Engineering at University of Hail, KSA. He is permanently Assistant Professor at Hodeidah University, Yemen. His research focuses on Wireless Networks, Mobile Networks, Vehicle networks, WSN, Cybersecurity, and Image Processing. He is an IEEE member, Member, IAENG member, and ASR member. In his research area, he has published many papers in reputed journals and conferences.

**Zeyad Ghaleb Al-Mekhlafi** received the B.Sc. degree in computer science from the University of Science and Technology, Yemen, in 2002, the M.Sc. degree in computer science from the Department of Communication Technology and Network, Universiti National Malaysia (UKM), in 2011, and the Ph.D. degree from the Department of Communication Technology and Network, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, in 2018. He is currently a Lecturer with the University of Ha’il, where he is also an Assistance Professor with the Faculty of Computer Science and Engineering. His current research interests include wireless sensor networks, energy management and control for wireless networks, time synchronization, bio-inspired mechanisms, and emerging wireless technologies standard.