Detection of Phishing Websites With Various Data Mining Classification Techniques

KOFI AGYEMAN, JOSIAH ANDERSON, NATHAN GOOTEE, POORVAJA SUNDAR, and CINDY WEI, University of California, Riverside

The internet today is irreplaceable. The amount of life, industry and commerce is immense. There are various applications that make life easier for people in many ways, from banking to online shopping. But alongside the many advantages, one of the primary disadvantages can be identified as phishing. There are many phishing websites whose aim is to trap users and extract sensitive information from them. The main aim of this paper is to identify legitimate websites versus phishing websites, and obtain the accuracy measures of these predictions. This is done using five well-known data mining algorithms: Support Vector Machine, Logistic Regression, Naive Bayes, AdaBoost Technique and Decision Trees. We determine which algorithms best classify the dataset considered and give good accuracy measurements.

CCS Concepts: • **Computing methodologies → Classification and regression trees**.

Additional Key Words and Phrases: phishing, classification, support vector machine, naïve bayes, logistic regression, adaptive boosting, decision tree

ACM Reference Format:
Kofi Agyeman, Josiah Anderson, Nathan Gootee, Poorvaja Sundar, and Cindy Wei. 2020. Detection of Phishing Websites With Various Data Mining Classification Techniques. 1, 1 (June 2020), 8 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

It is difficult to imagine life today as we know it without the Internet. Our societies depend upon it in a manifold number of ways, such that little thought is often given to whether the websites and applications we take for granted are indeed legitimate and safe. Web services are used for various activities like social media, banking, buying, selling and transferring money. Given the integrated and seamless nature of online activity, malicious actors frequently attempt to retrieve private and sensitive information from unaware users, which leads to serious problems for many of their victims. There are vast numbers of malicious websites on the Internet, despite great efforts to thwart them, and unsuspecting users are daily becoming victims to phishing, spam and drive-by-downloads. Phishing is defined as the social engineering processes which lure users into accessing, and often trusting, fraudulent websites which seek to obtain their personal or sensitive information, such as user names, passwords, addresses and credit card information. According to the Anti-Phishing Working Group (APWG) Phishing Trend Report, in the first quarter of 2020, a number of "cyber-criminals" have launched various COVID-19 themed phishing and malware attacks against employees, healthcare facilities and the recently unemployed. There was a significant increase in the number of phishing sites detected in the first quarter of 2020 when compared to the last quarter of 2019, an approximate increase of 3000 known phishing sites. A major challenge facing detection of phishing sites is the discovery of techniques being utilized by malicious actors. They continuously enhance their techniques and develop web pages that are able to withstand various forms of detection. Due to this, there is also a need to actively predict and classify phishing sites. The aim of this paper is to present a study of existing methods that are used to classify phishing sites versus legitimate sites. The dataset we employ was extracted from Kaggle and relatively found to be relatively clean with no missing values. This study uses algorithms such as Support Vector Machines, Decision Trees, Logistic Regression, Naïve Bayes and AdaBoost Classifier. We determine the accuracy of classification for each algorithm and reach conclusions as to which perform best.

2 RELATED WORK

2.1 A Comparison of Machine Learning Techniques for Phishing Detection [1]

This paper is concerned with the comparison of logistic regression, classification and regression trees, Bayesian additive regression trees, support vector machines, random Forests, and neural networks for predicting phishing emails.

The techniques used are well considered and optimized for the problem. The techniques used are generally implemented in a standard fashion, except for the Bayesian additive regression trees (BART). BART is originally designed for regression problems. The paper adapts BART for classification by approximating its quantitative output to a binary prediction via an experimentally determined threshold value.

The paper evaluates and compares the algorithms based on precision, recall, f1 scores, false positive rate, false negative rate and AUROC (area under receiver operating characteristic curve). Furthermore, weighted error is used to evaluate the algorithms effectiveness. This weighted error places a greater penalty on false positives, as the paper considers them to be more costly in real life scenarios.

The paper contributes a well-considered comparison and evaluation of various classification techniques for phishing detection. The advantages of this paper lie in the careful and reasoned implementation of classification techniques and evaluation scheme. Despite detailed comparisons and rankings of various metrics and parameter settings, the salient disadvantage of this paper is that there is no decisive conclusion on which method is superior. This is due to the subjective and theoretical elements of the evaluation scheme. A potential solution to this issue is the use of the AdaBoost technique to combine weak learners.

As in this paper, logistic regression and support vector machines will be evaluated in the present study for their effectiveness in

Authors' address: Kofi Agyeman; Josiah Anderson; Nathan Gootee; Poorvaja Sundar; Cindy Wei, University of California, Riverside, Riverside, California.

phishing website classification. Furthermore, the effectiveness of AdaBoost will also be investigated.

2.2 Predicting Phishing Websites using Classification Mining Techniques with Experimental Case Studies[2]

In the case study from Aburrous et al, they were using classification algorithms to predict phishing websites. They proposed a novel approach to use association rule mining through the Associative Classification techniques to predict phishing sites. It is a new concept because they paired association rules with the classification methods. This improves the accuracy of the classification mining techniques.

They used the data from two archives, the Anti Phishing Working Group (ATWG) and Phishtank.com, in which the data were collected between 2007 and 2008. The authors came up with 27 phishing attributes by conducting a case study with employees in the Jordan Ahli Bank. Next, these attributes were grouped into six categories, which then serves as the input class attributes for the Associative Classification.

For the six data mining association and classification techniques examined, C4.5, RIPPER, PART, and PRISM were evaluated using WEKA software, which is a java open-source platform. The Association Classification techniques, MCAR, and CBA follow the same method used by the presentation from Liu et al., 1998.

The Associative Classification outputs a class attribute based on the input items. Each attribute of the six phishing categories was placed in one of the fuzzy set values - legitimate, doubtful, and genuine, which was defined in a previous paper written by the authors. The evaluation included the prediction accuracy rate and the number of rules generation.

The Associative Classification technique looks promising with higher accuracy than the traditional classification algorithms. it would be interesting to compare associative classification with a hybrid approach. The hybrid approach meaning pairing two traditional classification algorithms together.

The advantage of the Associative Classification was that you can examine the correlation of the attributes. The experiment found that Associative classifiers improve the model with higher accuracy than the traditional classification algorithms.

2.3 Phishing Detection Based on Machine Learning and Feature Selection Methods[4]

The paper is about presenting a study of existing methods used in detection of phishing web pages that employ machine learning algorithms and focus on the most common feature selection methods. The paper then goes on to compare the performance of the different machine learning algorithms such as J48, Random Forest and Multilayer Perceptron and find out which one is more efficient. 48 is a type of C4.5 decision tree algorithm. The Random Forest is a classification method based on the decision tree algorithm. A Multilayer Perceptron is the most popular and frequently used artificial neural network. The feature selection methods are employed to decrease the size of the data set and also improve the performance of the model and decrease the computation time. The methods employed by the paper are: InfoGain, ReliefF. The evaluation method used in this paper is the accuracy equation because the dataset used in both binary and balanced. Accuracy is the ratio of the sum of true negative and true position and the total population. The best results obtained after evaluation is that the Random Forest algorithm performed best with the time taken to construct the model is 2.44 seconds and accuracy rate of 98.11, when the features are reduced to 20 from 48. The paper concludes that the accuracy is at 98.11, but we would like to use more number of machine learning algorithms to cross check if Random Forest is indeed the best algorithm.

2.4 Phishing websites classification using hybrid SVM and KNN approach[5]

This paper is trying to detect phishing websites using a hybrid approach of combining the advantages of KNN with SVM. They selected the dataset prepared by another paper and compared the performance of the hybrid approach with basic SVM, Naive Bayes, Neural Network, Decision Tree, and basic KNN. The effectiveness of the performances was evaluated by two metrics: accuracy and recall. Additionally, the performance of the hybrid approach was also compared with related and existing approaches done by other studies.

As the accuracy and recall metrics and the related existing approaches both conclude the hybrid approach had a higher accuracy. This supports the potential of the hybrid approach as an improvement to the existing algorithms.

The pros of doing a hybrid approach were reducing the shortcomings of the individual algorithm. Depending on the combination of techniques selected, it can be computationally expensive if the algorithm goes through every data point. Conversely, it can save time and money with efficient algorithms.

A potential expansion on the paper is to experiment with pairing different levels of techniques to better understand the effectiveness of the combination. For instance, examine the combination of a simple classification algorithm with a less complex algorithm. Another one would be pairing a simple technique with an advanced algorithm. One more could be combining two advanced algorithms together.

2.5 Malicious-URL Detection using Logistic Regression Technique[10]

This paper by Vanitha and Vinodhini attempts to solve phishing website classification using logistic regression. This paper has significant shortcomings yet offers a solid overview on phishing classification with logistic regression, and nuance in some aspects of solving this problem. The implementation of the logistic regression algorithm is basic, the researchers used the scikit learn algorithm and did not indicate that hyperparameters were modified in any way. This is an obvious disadvantage to the paper's approach to phishing classification.

Nuance is introduced with the use of the TF-IDF vectorizer algorithm to generate features. The original data is only website URLs with labels to indicate if the site is a phishing site or not. The TF-IDF vectorizer is then used to generate data based on the lexical features of the URLs such as length and domain name entropy. This is an advantage to the paper's approach as researchers would have a stronger understanding on why data points have certain values for each feature. The value of the paper's results is affected by assumptions that were made throughout. There is not enough information given on the use of the TF-IDF vectorizer and generated features. Consequently, it is unknown if their data was normalized or not. As previously mentioned, the paper does not discuss optimization of hyperparameters. The paper therefore assumes that the default hyperparameter settings are optimal. Furthermore, it is not clear how reliable the original data source was. The paper assumes that the URLs gathered from the third party (a currently inactive GitHub repository) were labelled correctly (phishing/safe), without verification or evaluation. Lastly, the paper assumes that the training data has a similar ratio of true/false labels as the test data. This assumption is made as the paper is not explicit about stratification during data splitting.

The paper ultimately asserts that logistic regression is a superior algorithm for phishing classification in comparison to naïve Bayes and random forest algorithms, by the metric of algorithm accuracy. Several extensions of this paper could be explored. The researchers should consider use of a validation set or cross validation, hyperparameter optimization and other evaluation metrics such as f-1 score, AUROC, sensitivity and specificity. This paper provides a limited insight into the effectiveness of logistic regression in this problem space.

2.6 Detection of Phishing URLs using Machine Learning Techniques[6]

The aim of the paper is to derive classification models that detect phishing web sites by analysis of the lexical and host-based features of URLs. The classifying algorithms are analysed in WEKA workbench and MATLAB. The paper extracts the host-based, page-based and lexical features of collected URLs and analyses it. The design flow is as follows: 1. Collect phishing and benign URLs 2. Host-based and page-based feature extraction 3. Lexical feature extraction 4. Evaluate features using machine learning algorithms 5. Selection of Classifier 6. Implement the Classifier. The machine learning algorithms used are : 1. Naïve Bayes 2. J48 Decision Tree 3. K-Nearest Neighbours 4. Support Vector Machine. The findings of the preliminary work include: 1.Phishing URLs and domains exhibit characteristics that are different from other URLs and domains. 2. Phishing URLs and domain names have quite different lengths compared to other URLs and domain names in the Internet. 3. Many of the phishing URLs contained the name of the brand they targeted. The URL feature dataset was split accordingly: 40% as the training data and 60% as the test data. The analysis of the dataset is then done using MATLAB and WEKA. There are slight differences in the values of Success Rate provided by MATLAB and WEKA, and the Decision Tree is found to have the highest success rates as compared to the other algorithms. When the percentage split is 60 in WEKA, J48 provides a success rate of 93.20, and with a percentage split of 90, it gives a success rate of 93.78. Likewise, in MATLAB, the regression tree algorithm for percentage split 60 gives 91.08 success rate and 85.63 success rate with percentage split as 90. The main drawback of the paper is that there is not a clear explanation for the extraction of lexical features when compared to host-based and page-based. Also, there is the case of the existence of new websites, which have the PageRank as Not Available. This could easily be mistaken as a phishing website. In our paper, we are trying to observe if we get better accuracy measure results with the algorithms already used above gives a different result for a different data set.

2.7 Intelligent web-phishing detection and protection scheme using integrated features of Images, frames and text[3]

This paper published in 2019 proposed the use of related features of images, frames and text from legitimate and non-legitimate websites and associated artificial intelligence algorithms to develop an integrated phishing detection and protection scheme to prevent phishing attacks aimed at enticing victims to divulge confidential and sensitive information. The proposed scheme combines a novel neural network and fuzzy logic algorithm using a total of 35 features; 22 text, 8 frame and 5 image-based resource elements, and ANFIS to differentiate between legitimate, suspicious and phishing websites. ANFIS, a network structure method that utilizes a fusion learning technique to combine least squares and gradient descent methods to derive output errors and derive fuzzy IF… THEN rules.

According to the researchers, the detection system structure is composed of five parts: website analysis and features extraction, intelligent system, knowledge model, knowledge sources and output process. They assume text features only machine learning phishing detection approaches are less effective. The main contribution of the paper according to the researchers is the hybrid use of only website frames, images, and text features to detect phishing websites automatically in real time that improves previous work that used text-based features only. Their approach also explores computer vision techniques for detection.

Datasets from UC Irvine and University of Huddersfield with approximately 13000 datasets with equal features weights assignment (-1, 0, 1) was utilized to test and evaluate the scheme with 4898 phishing, 1945 suspicious and 6157 legitimate websites. The paper's novel ANFIS phishing detection approach performance was compared to conventional SVM and K-NN algorithm performances using confusion matrix to calculate accuracy, precision, recall and F-Measure measures. ANFIS performed with performance accuracy of 98.55% using text only features and 98.30% utilizing the hybrid text, image and frames features.

For future work the group intends to include deep learning algorithms and a web browser plug-in to improve on efficiency and integration.

2.8 On Designing and Evaluating Phishing Webpage Detection Techniques for the Real World[7]

There is an observed gap between the high accuracy reported by phishing detection techniques in the literature and the actual dramatic success rates of phishing attacks. This paper claims there are two major reasons: 1) Often the only criterion in the design phase is detection accuracy levels. 2) Evaluation methodology is not representative of real-world scenarios. The authors contend that ground truth collection, phishing dataset selection and sanitization, dataset usage, accuracy metrics, temporal resilience and longitudinal studies are critical components that are often overlooked.

2.9 New Rule-Based Phishing Detection Method [8]

This paper proposed to use two novel rule-based feature sets to improve the performance of detecting phishing attacks and prevent data loss in internet banking. The group aimed to extend the general support vector machine (SVM) learning classification approach for detecting phishing attacks through analysis of website appearance information or web address characteristics by determining the relationship between web content and page URL.

According to the paper, their method uses four additional features to evaluate page resource identity and another four features to identify the access protocol to page resource elements, all related to page links, JavaScript files, style sheet files and images. Using knowledge extracted from a C4.5 classifier rule induction algorithm from the classification step, they use a web browser extension to implement their 10 rules rule-based detection method.

One pro of this rule-based knowledge extraction approach, they claim, is increased comprehensibility of the logic behind the SVM model approach which essentially is a "black box". Other advantages include independence of the scheme on third-party services such as search engines and/or black/white site lists, or on a websites' language because the model features are extracted from page content. Further, the proposed method can detect zero-day phishing attacks. However, the use of web page content only for detection assumes that most attackers do not redesign phishing webpages or that the web content is not an image of flash media, in which case extraction of features from web page DOM would fail. Other downsides of the approach are that it requires user intervention to execute the embedded extension and it is browser specific; works with Google Chrome browser only.

To train the model and evaluate its performance, a stratified sampling scheme was used to select legitimate webpages data from Yahoo directory service and phishing webpage data from PhishTank. 17 total features were used. Confusion matrix, Kappa statistic, sensitivity and F-score metrics were used for evaluation of 3 feature sets; each composed from literature, proposed features and combined. Individual feature sensitivity analysis was also performed to assess the importance of each of the 17 features used. Overall, the combined feature set performed best with TP of 99.14% and FN rate of 0.86%.

In terms of future directions, the use of heuristics approach in selecting and defining the additional features can be improved. A systematic feature selection approach using techniques such as principal component analysis (PCA) to select most significant features may improve their accuracy, sensitivity, and specificity of the method or to confirm that the optimum feature set is selected.

2.10 Intelligent Rule-Based Phishing Websites Classification[9]

This paper proposes using rule-based classification of extracted features to determine the "phishiness" of websites, via real-time heuristic-based methods. Data mining techniques present themselves as indispensable in this regard. Four algorithms were compared for performance: C4.5, RIPPER, PRISM and CBA. The most predictive features were determined to be: Request URL, Age of Domain Name, HTTPS Issuer / Age, Website Traffic, Long URL, Subdomain and Multi-Subdomains, Domain Prefix/Suffix Hyphenated, URL of Anchor, and Using an IP Address in URL.

3 PROPOSED METHODS

3.1 Support Vector Machine

Support vector machines (SVM) linear classifier model was developed and utilized to learn a predictive classification model for phishing detections. Dataset courtesy of Kaggle data repository was loaded, inspected, and partitioned into training and testing (20%) predictors array and output series. See Fig. 1 for details of the data break down. There were no missing values and all predictor feature values were either [-1, 0, or 1], thus mitigating the need for min/max or Gaussian normalization of the feature values. Linear SVM model training was implemented with assessment of various cost thresholds and learning rates using partitioned data. This was followed by optimization analysis investigating SVM feature correlation-based selection and linear, polynomial, and Gaussian kernels performance. Kernel and feature selection analysis were performed with "sci-kit learn" functions. Correlation analysis was used to attempt to reduce the number of features by eliminating attributes with high correlation.

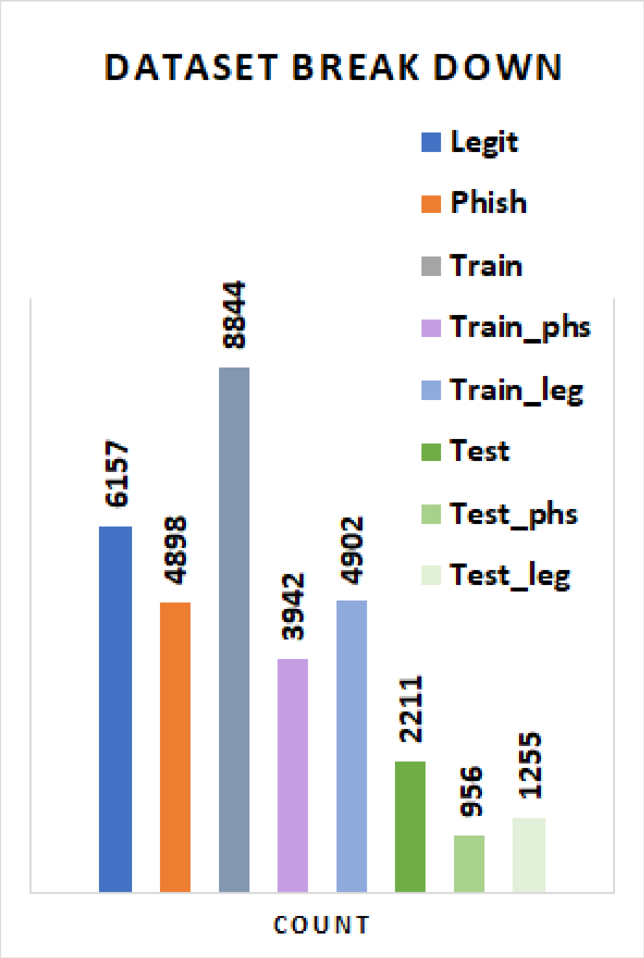


Fig. 1. Data break down

3.2 Naïve Bayes

Naïve Bayesian classifiers are known for their speed and computational simplicity. Naïve Bayes is a probabilistic classifier assuming conditional independence, and is well suited for detecting phishing websites. Our implementation here is a Bernoulli Naive Bayes, due to the binary nature of the features. We choose to ignore any "zero" attributes from the tuple, in part because of the multiplication of the probability of features which occurs in Bayesian computations cannot compute zero values, as multiplying anything by zero is zero. Instead, a Laplacian-inspired "smoothing" value of 0.0000000000001, is inserted in its place. Values lower than this were found to significantly decrease classification accuracy. There was one column in our dataset which did not follow the -1, 0, 1 range, "redirect". This column was eliminated as part of the data preprocessing for the Bayes implementation.

Our data set was split into 80% training data and the rest as test data. Our training data is then grouped into two classes, "phishing" and "legitimate", derived from the labels in the "result" column. Class probabilities are calculated by dividing the number of legitimate and phishing sites by the total number of sites. Feature probabilities are computed by finding the probability of each feature given the class label. Classification of the test data is performed by iterating through each row, and calculating the probability of each feature for legitimate and phishing classifications. The product of all features of the row is then multiplied by the class prior probability derived from the training data. Laplacian correction is provided here to ensure that none of our feature probabilities are zero. The two calculated probabilities, phishing and legitimate, are then compared and the higher probability is assigned the respective class label of "phishing" or "legitimate". Our classification labels are then passed to the Python sklearn metrics library, along with the truth data labels from the test set.

3.3 Logistic Regression

Logistic Regression is a classification algorithm that typically utilizes the sigmoid function for binary predictions, though it can be modified for multinomial classification. The algorithm is implemented for binary output as phishing detection is a binary classification problem.

The inputs are weighted and then fit to the sigmoid function to provide a hypothesis for classification as shown in the following equation:

$$hypothesis = \frac{1}{1 + e^{-\theta X}}$$

Where X is the matrix of observations for input features and θ is the weight given to each feature in X. During training, weights are evaluated through a cost function which computes the accuracy of each point's prediction given current weights. This cost function is the inverse of the log likelihood function for logistic regression where cost is given by:

$$cost = -Y \log(hypothesis) - (1 - Y) \log(1 - hypothesis)$$

Where Y is an array of the observation labels (safe/phishing). The function returns the mean cost for each data point. Gradient descent is used in training to minimize this cost function and set the optimal weight of each feature. Gradient descent was chosen over other methods for its simplicity.

The data is split so that 80% is used for training and 20% for test. The split is stratified through labels so that the ratio of the labels is preserved in both data sets. When fitting the training set, the learning rate and convergence value for gradient descent must be specified. These parameters are set at 0.0001 and 0.0000001 as these setting produced the best accuracy. Fitting the training set returns the vector of weights and number of steps that the gradient descent completed before reaching convergence. Predictions for the testing data set requires the previously determined feature weights to be input. The predict function then returns a vector of predicted labels for the testing set.

Logistic Regression is often praised for its simplicity, interpretability and efficiency. The algorithm does have its disadvantages though. Logistic Regression has poor performance when there are nonlinear relationships between features and class. Furthermore, Logistic Regression relies on certain statistical assumptions, such as that all features are conditionally independent given the class label. Through exploration of the data set it was determined that feature/class relationships were linear and that features are conditionally independent given the class label.

Some minor challenges were encountered during implementation of logistic regression. Firstly, class labels needed to be altered from 1(legitimate) and -1(phishing) to 1(legitimate) and 0(phishing) due to the bounds of the sigmoid function. This was a simple adjustment. Some issues arose with matrix operations, this was fixed through the NumPy squeeze function which removes dimensions of magnitude 1. This operation is also applied to the prediction output, so that the output and input are in the same format for comparison and evaluation purposes. Furthermore, different subsets of input features were examined for their effect on accuracy and speed. Ultimately, accuracy was best with all input variables used, with minimal effect on speed.

3.4 Decision Tree

Decision tree is a prevalent model due to its ease of understanding for humans and it forces a deterministic outcome. This greedy algorithm can have low accuracy due to its high bias in multivalued attributes. The dataset is relatively small compared to the gigabytes of data and the number of features of 31 is finitely manageable. Our dataset is all numeric and the labels are deterministic (1=phish website and -1= not a phishing website), so CART (a binary decision tree) is well suited for this problem.

I split the dataset into 80/20 for training and test. My Gini index is set to Gini(dataset) = 1 - the sum of Pj2, where Pj is the relative frequency of class j in the dataset.

3.5 Adaptive Boosting

The adaptive boosting algorithm is one which creates a strong classifier from a number of weak classifiers by learning from the incorrect predictions of weak classifiers. In the dataset chosen, the number of classifiers chosen is 4, and the dataset is split such that 20% of the dataset is test data and the remaining is training data. The chosen weak learner is Decision Trees and each classifier has a depth of 1. After the dataset is split, the AdaBoostClassif class is called, and the fit and predict functions are called.

In the fit function, the initial weights are assigned as $1/n$ where n is the number of samples in the dataset. Then the dataset is applied to the first classifier and the minimum error is found. After the minimum error is calculated, the performance of the classifier is calculated using

$$output = \frac{1}{2}\log(\frac{1 - error_{minimum}}{error_{minimum}})$$

where log is of base e. The weights are then recalculated by multiplying weight and the exponent of performance of the classifier.

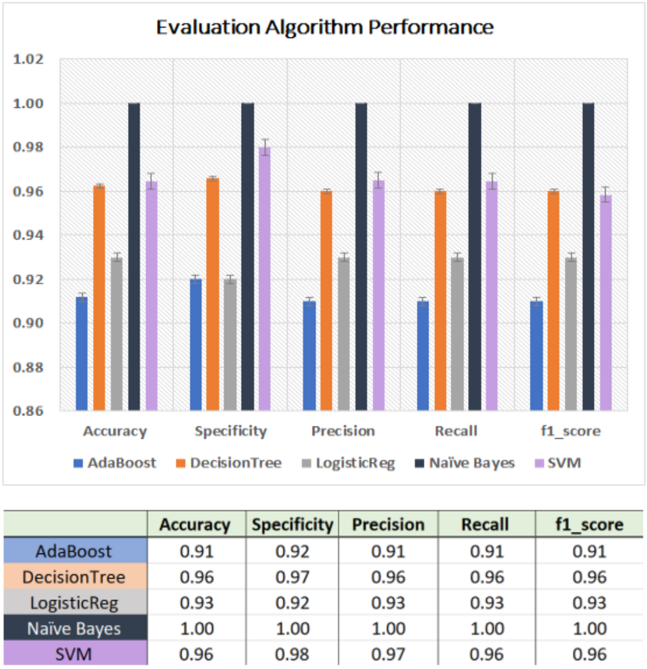
$$weight = weight * e^{-output}$$

We need to make sure that the sum of weights is 1. So we sum the weights and divide the weight of each sample with the sum. We then combine all the weights measured by the weak learners into a weight of a strong learner. This weight is then used in the prediction of whether a website is a legitimate or illegitimate website.

The predict function is then called to label the train data as a valid website or a phished website based on the weights measured in the fit function. Based on the predictions in the train data, the training data is passed into the algorithm to obtain the accuracy.

4 EXPERIMENTAL EVALUATION

By comparing the obtained accuracy measures in each algorithm, it is observed that Naive Bayes is by far the best performing algorithm for the chosen dataset. The Naive Bayes algorithm gives an accuracy measure of 99%, which is the best accuracy measure when compared to the others. Figure 2 depicts the comparison of accuracy, precision, recall and f1-score of all the five algorithms in the form of a bar chart. This chart greatly helps in the observation of the best algorithm with ease

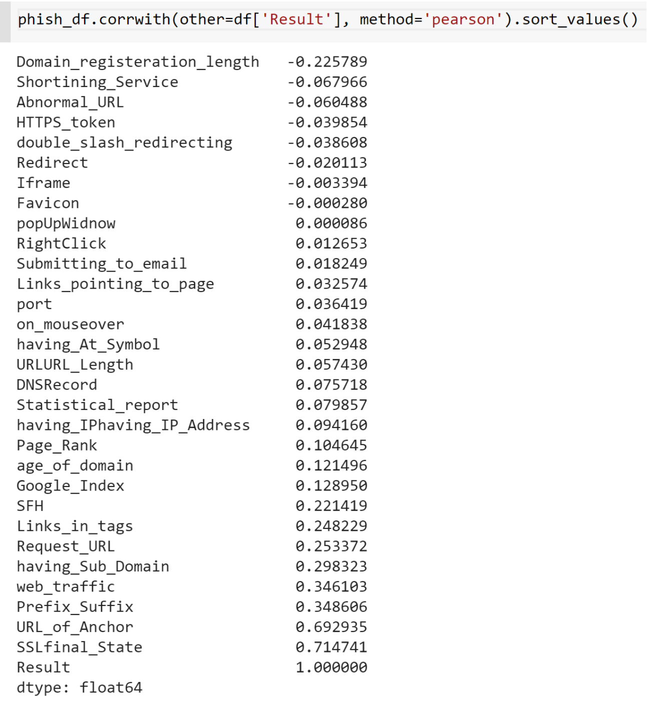


	Accuracy	Specificity	Precision	Recall	f1_score
AdaBoost	0.91	0.92	0.91	0.91	0.91
DecisionTree	0.96	0.97	0.96	0.96	0.96
LogisticReg	0.93	0.92	0.93	0.93	0.93
Naïve Bayes	1.00	1.00	1.00	1.00	1.00
SVM	0.96	0.98	0.97	0.96	0.96

Fig. 2. Graphic evaluation of results

5 DISCUSSION AND CONCLUSIONS

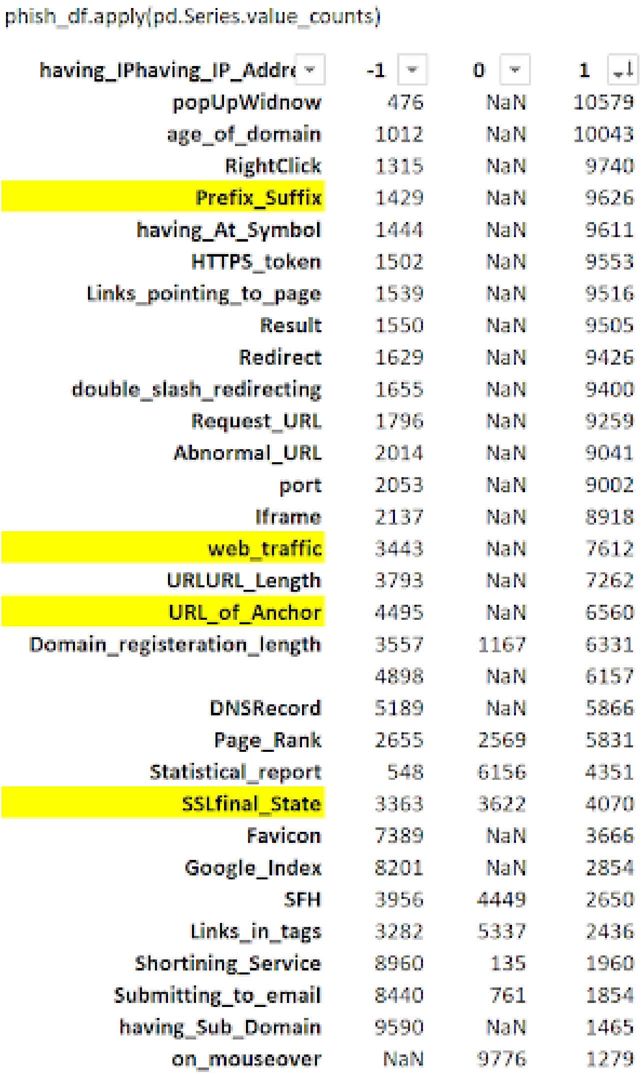
The selected phishing dataset is very clean in that there are no missing values, attributes' definitions are made available and the labels are deterministic. High data quality is ideal because that trim down the pre and post-processing time, as well as reduce the complexity of the classifier. Generally, there are apparent disparities in the results from various classifiers and in our case, the five algorithms all performed really well with accuracy well above 90%. A clean and high-quality dataset is a huge factor and contributed to this result. Taking a look into the features to see if there's any correlation. The top features correlated with phishing sites are Prefix_Suffix, web_traffic, URL_of_Anchor, and SSLfinal_State. (Fig 3). When we look at those same features (Fig 4) laid out sorted by its values, the Prefix_Suffix is high on top of the list. This seems to suggest that Prefix_Suffix has a high correlation and it may be a good indicator to detect phishing websites as Mohammad et al. have concluded the similar in their experiment [9].



```
phish_df.corrwith(other=df['Result'], method='pearson').sort_values()

Domain_registeration_length   -0.225789
Shortining_Service            -0.067966
Abnormal_URL                  -0.060488
HTTPS_token                   -0.039854
double_slash_redirecting      -0.038608
Redirect                      -0.020113
Iframe                        -0.003394
Favicon                       -0.000280
popUpWidnow                    0.000086
RightClick                     0.012653
Submitting_to_email            0.018249
Links_pointing_to_page         0.032574
port                           0.036419
on_mouseover                   0.041838
having_At_Symbol               0.052948
URLURL_Length                  0.057430
DNSRecord                      0.075718
Statistical_report             0.079857
having_IPhaving_IP_Address     0.094160
Page_Rank                      0.104645
age_of_domain                  0.121496
Google_Index                   0.128950
SFH                            0.221419
Links_in_tags                  0.248229
Request_URL                    0.253372
having_Sub_Domain              0.298323
web_traffic                    0.346103
Prefix_Suffix                  0.348606
URL_of_Anchor                  0.692935
SSLfinal_State                 0.714741
Result                         1.000000
dtype: float64
```

Fig. 3. Correlation between features relative to the Label



phish_df.apply(pd.Series.value_counts)

having_IPhaving_IP_Addre	-1	0	1
popUpWidnow	476	NaN	10579
age_of_domain	1012	NaN	10043
RightClick	1315	NaN	9740
Prefix_Suffix	1429	NaN	9626
having_At_Symbol	1444	NaN	9611
HTTPS_token	1502	NaN	9553
Links_pointing_to_page	1539	NaN	9516
Result	1550	NaN	9505
Redirect	1629	NaN	9426
double_slash_redirecting	1655	NaN	9400
Request_URL	1796	NaN	9259
Abnormal_URL	2014	NaN	9041
port	2053	NaN	9002
Iframe	2137	NaN	8918
web_traffic	3443	NaN	7612
URLURL_Length	3793	NaN	7262
URL_of_Anchor	4495	NaN	6560
Domain_registeration_length	3557	1167	6331
	4898	NaN	6157
DNSRecord	5189	NaN	5866
Page_Rank	2655	2569	5831
Statistical_report	548	6156	4351
SSLfinal_State	3363	3622	4070
Favicon	7389	NaN	3666
Google_Index	8201	NaN	2854
SFH	3956	4449	2650
Links_in_tags	3282	5337	2436
Shortining_Service	8960	135	1960
Submitting_to_email	8440	761	1854
having_Sub_Domain	9590	NaN	1465
on_mouseover	NaN	9776	1279

Fig. 4. Features' value classification in ascending phishing order

REFERENCES

[1] Saeed Abu-Nimeh, Dario Nappa, Xinlei Wang, and Suku Nair. 2007. A comparison of machine learning techniques for phishing detection. *Proceedings of the anti-phishing working groups 2nd annual eCrime researchers summit on - eCrime* (Oct. 2007). https://doi.org/10.1145/1299015.1299021

[2] Maher Aburrous, M. A. Hossain, Keshav Dahal, and Fadi Thabtah. [n.d.]. Predicting Phishing Websites using Classification Mining Techniques with Experimental Case Studies. *2010 Seventh International Conference on Information Technology: New Generations* ([n. d.]). https://doi.org/10.1109/itng.2010.117

[3] M. A. Adebowale, K. T. Lwin, E. Sánchez, and M. A. Hossain. 2019. Intelligent web-phishing detection and protection scheme using integrated features of Images frames and text. *Expert Syst. Appl.* (Jan. 2019). https://doi.org/10.1016/j.eswa.2018.07.067

[4] Mohammad Almseidin, AlMaha Abu Zuraiq, Mouhammd Al-kasassbeh, and Nidal Alnidami. 2019. Phishing Detection Based on Machine Learning and Feature Selection Methods. *International Journal of Interactive Mobile Technologies (iJIM)* (2019). https://doi.org/13.171.10.3991/ijim.v13i12.11411

[5] Altyeb Altaher. 2017. Phishing Websites Classification using Hybrid SVM and KNN Approach. *IJACSA* (2017). https://doi.org/10.14569/ijacsa.2017.080611

[6] Joby James, L. Sandhya, and Ciza Thomas. 2013. Detection of phishing URLs using machine learning techniques. *2013 International Conference on Control Communication and Computing (ICCC)* (Dec. 2013). https://doi.org/10.1109/ICCC.2013.6731669

[7] Samuel Marchal and N. Asokan. 2018. On Designing and Evaluating Phishing Webpage Detection Techniques for the Real World. *CSET '18: 11th USENIX Workshop on Cyber Security Experimentation and Test.* (2018). https://www.usenix.org/system/files/conference/cset18/cset18-paper-marchal.pdf

[8] Mahmood Moghimi and Ali Yazdian Varjani. 2016. New rule-based phishing detection method. *Expert Systems with Applications* (July 2016). https://doi.org/53.10.1016/j.eswa.2016.01.028

[9] Rami M. Mohammad, Fadi Thabtah, and Lee McCluskey. 2013. Intelligent rule-based phishing websites classification. *IET Information Security* (July 2013). https://doi.org/10.1049/iet-ifs.2013.0202

[10] Vanitha N and Vinodhini V. 2019. Malicious-URL Detection using Logistic Regression Technique. *International Journal of Engineering Business Management* (Dec. 2019). https://doi.org/108-113.10.31033/ijemr