Chapter 11
An Efficient Approach for Phishing Detection using Machine Learning

Ekta Gandotra and Deepak Gupta

1 Introduction

Due to the availability of Internet at low price, people are shifting to online platforms instead of visiting banks, shops, etc. Attackers are taking advantage of this fact and trying to find their victims online to make money instead of taking risks to rob banks/shops, etc. They are making use of various attacks like phishing to steal the passwords, credit card details, etc., by misleading users to visit malicious and fake websites.

Phishing attack is one of the top security threats on the Internet today. Attackers tend to gather victims' confidential information using fake websites. According to Anti-Phishing Working Group [1], the trend of phishing attacks is increasing every year and 138,328 phishing pages were informed in 2018, 4th quarter. It causes a lot of financial losses. On the basis of the cases informed to Federal of Investigation [2], there occurred a loss of around $48 million in USA in the year 2018. In addition, phishing attacks are also becoming the top delivery method of malware [3–5]. A recent report of Microsoft security intelligence [6] reported that in 2018, phishing attack was the top web attack.

In year 2013, South Korea faced a phishing attack, in which 3 banks and 3 media companies were compromised [7]. This attack involved stealing passwords and was spread to more than 32,000 machines. In December 2014 [8], the cyber attackers were able to get the data of 80 million customers of Anthem Healthcare. The recent

E. Gandotra
Department of Computer Science and Engineering, Jaypee University of Information Technology, Waknaghat, Solan, India
e-mail: ekta.gandotra@gmail.com

D. Gupta (✉)
Department of Computer Science and Engineering, Thapar Institute of Engineering & Technology (Deemed to be University), Patiala, India
e-mail: deepak.vd@gmail.com

K. J. Giri et al. (eds.), *Multimedia Security*, Algorithms for Intelligent Systems,
https://doi.org/10.1007/978-981-15-8711-5_12

worldwide lockdown due to novel coronavirus (COVID-19) epidemic has forced many employees to work from home where they do not have cybersecurity resources as they usually have in their offices. Moreover, due to the infectious nature of the virus, there is anxiety among the people over any online information related to coronavirus. The attackers are exploiting these facts to perform phishing attacks through emails or fake websites with the aim of making money. Threat Analysis Group [9] of Google has identified attacker groups who are using subject as COVID-19 to attempt malware and phishing attacks.

There exist various tools [10–12] which attackers use to duplicate a website in order to create a phishing website. Sometimes, the websites designed in such a way are identified by anti-phishing tools because of the website's elements such as copyright, anchor link, logo or keywords of the genuine websites. There are various approaches for detection of phishing websites which are having their own pros and cons. Signature-based method is the most popular method used for detecting phishing webpages. This method maintains a database of blacklisted websites which consists of a list of phishing webpages. Any newly requested webpage is checked against this database to identify whether it is phished. Two of the important blacklisted databases are PhishTank [13] and Google Safe Browsing (GSB) [14]. This approach fails to identify the newly generated phishing websites which have not been added to the database. In case of heuristic approach, the webpage is analyzed to come up with discriminating features like process creation, page redirection, etc. [15]. However, this approach is resource-intensive. Another approach used to detect phishing websites is visual-based approach [16] in which a database of images, logos, text formatting, HTML tags, etc., is created. But this method is highly time-consuming.

Recently, researchers have started to employ machine learning and computational [17] based methods for identifying phishing websites. In this method, different types of features pertaining to phished and benign websites are extracted, and subsequently, these are used for training purpose to build the classification models. The efficiency and performance of such methods depend on the classification algorithms, number of training examples, and the feature set. Keeping the classification models and the number of training examples apart, considering a large and variety of features helps in enhancing the performance of classification models. However, it takes more time for model building which hampers the timely detection of phishing websites. Therefore, an appropriate set of features should be selected to build the classification models in less time without compromising the accuracy. Keeping this point in mind, this chapter aims to present a study on the role of feature selection methods in detecting phishing webpages. Consequently, a comparative analysis is performed on the performance of machine learning algorithms without and with feature selection.

To carry out the study, this chapter is structured as follows. Section 2 presents the recent research work in classifying the phishing webpages using machine learning techniques. Section 3 explains the methodology used in carrying out this study. Section 4 analyzes and discusses the experimental results. Section 5 provides conclusion and future scope.

2 Related Work

This section presents the recent research work pertaining to classification of phishing websites using machine learning algorithms.

Tan et al. [18] presented a method named PhishWHO for the detection of phishing websites in three stages. In first stage, the keywords are extracted from the websites (using N-gram method). In second stage, these keywords are used in a search engine to find the name of target domain. Finally, in the last stage, they used a matching system for detecting the legality of the website.

An image-based approach was proposed by Chiew et al. [19] for the detection of malicious websites. They extracted the logo and fed it into Google image search engine to check the identity of the website. They compared the webpage query with the domain name returned by Google for distinguishing a phishing webpage from the benign one. Experiments were conducted to prove the usefulness of the proposed method.

In [20], the authors analyzed various features of phishing webpages and shortlisted the most important 19 features. After extracting these 19 features from the source code of webpage, they used support vector machine (SVM), random forest (RF), neural network, logistic regression, and Naïve Bayes (NB) for classification of phishing webpages and obtained around 99% accuracy.

Rao and Pais [21] proposed a method which is based on human behavior while they get exposed to malicious webpages. This human behavior of feeding the fake credentials is automated with an additional step of applying heuristic filtering. They were able to get an accuracy of 96.38% using this approach.

In [22], the authors proposed a real-time system that makes use of the features pertaining to natural language processing (NLP). They used seven different machine learning algorithms for the classification of phishing and benign webpages. They tested their proposed system on a self-created dataset. Their results showed that RF with features based on NLP provides the best accuracy while classifying the webpages.

Xiaoqing et al. [23] presented an automatic intelligent system for the detection of phishing webpages. They analyzed the features of uniform resource locator (URL) and used NB for classification. In case of suspicious webpages, it is parsed and re-classified by using SVM. Through their results, they claim that the system gives high accuracy of detection in less time.

A similar approach was used in [24]. The authors presented a method which used a combination of SVM and decision tree model. SVM is used for training purpose and decision tree is used to generate the rules for detecting phishing websites targeting the banking domain.

“Cantina + ” [25] approach was used as an extension of “Cantina” [26] (based on textual content of a webpage and term frequency-inverse document frequency algorithm) where additional features were used to detect the phishing webpages. In this approach, the authors used eight features. They used search engines, document object model, and the services from the third party along with machine learning algorithms for the detection of phishing webpages.

Joshi et al. [27] proposed a system which selected important URL features and machine learning algorithms for the detection and classification of phishing webpages. They concluded that ReliefF and RF algorithms worked well than other combinations.

Wu et al. [28] presented a phishing detection tool that is developed by combining the URL of the webpage and its source code. They used Levenshtein method for computing the string similarity and SVM as a machine learning model in their proposed system for detection of phishing webpages.

Zamir et al. [29] analyzed the features of phishing data by using features selection methods. They proposed two features by combining various attributes. They used various machine learning and stacking techniques for detecting phishing webpages.

Almseidin et al. [30] presented a study where they employed various machine learning algorithms and features selection methods to improve the efficiency of their system. The experiments conducted on a phishing dataset of 48 features containing 5000 benign and 5000 phishing webpages. They concluded that RF algorithm with only 20 features gives the best accuracy.

Yerima and Alzaylaee [31] presented an approach based on deep learning for detecting phishing webpages. They used convolutional neural networks (CNN) for this purpose and evaluated on a dataset consisting of 6,157 genuine and 4,898 phishing websites. They compared their results with traditional machine learning algorithms and concluded that their approach using CNN gives much better results than conventional machine learning algorithms.

Basnet and Doleck [32] proposed a heuristic approach using URL-based features. Experiments were carried out on a dataset containing 138 features which were extracted from 31,000 malicious and 16,000 phishing webpages. These 138 features belong to four groups, namely search engine, reputation, lexical, and keyword-based. They used seven different machine learning algorithms for classification purpose. Random forest achieved the best accuracy.

Recently, due to the growth of multimedia systems [33, 34], there is an exponential growth in cyber-attacks. In order to protect the digital media, there is a need to develop the techniques to prevent their unauthorized distribution [35]. These involve digital signatures and water marking. A lot of research is being reported by the researchers on authentication of the content [36, 37].

From the literature, it is found that the research community has worked on improving the classification accuracy by considering a variety of large number of features for the classification of phishing webpages. However, it seems that there is less focus on the problem related to the building time of any classification model without compromising the accuracy. This chapter offers a study which compares the classification performance and efficiency of detecting phishing webpages using various machine learning algorithms without and with feature selection. It makes use of information gain method for selecting an appropriate set of features.

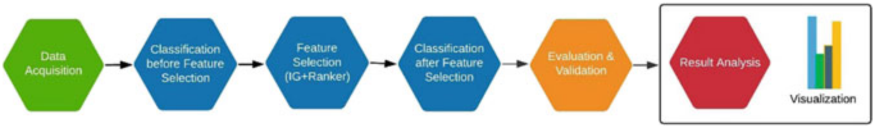
Data Acquisition
Classification before Feature Selection
Feature Selection (IG+Ranker)
Classification after Feature Selection
Evaluation & Validation
Result Analysis
Visualization

Fig. 1 Workflow of methodology used for proposed phishing detection system

3 Methodology Used

This section provides a description of the methodology used for discriminating benign webpages from the malicious ones. Figure 1 demonstrates the proposed design of phishing detection system. After downloading a suitable dataset, eight machine learning-based algorithms are employed to classify the phishing webpages. Afterward a feature selection method is deployed to choose top rank features. These selected features are then deployed to classify the webpages. Subsequently, the experimental results are compared by considering building time and performance of the models—before and after feature selection. The detailed steps of the proposed system are described in the following sub-sections.

3.1 Data Acquisition

Dataset used in this study is downloaded from [38, 39]. It is composed of 30 features (other than index and class label) extracted from 6157 legitimate and 4898 phishing webpages. Benign webpages are collected from Alexa [40] while phishing webpages are taken from PhishTank. This dataset is chosen in the present study because it is the most recent dataset available in the public domain.

3.2 Classification before Feature Selection

In this work, we have used a free data mining tool—Waikato Environment for Knowledge Analysis (WEKA), version 3.8.4 for Windows Operating System [41]. We used eight machine learning classifiers, namely IB1, NB, J48, AdaBoost, decision table (DT), bagging, RF, and sequential minimal optimization (SMO) for classifying phishing webpages. In this step, all 30 features present in the original dataset are used for constructing the classification models. A brief description of these algorithms is given below:

- IB1 is a classifier that stores the training data and starts building the classification model only when it gets the test data. Due to this reason, it is also known as lazy learner. It is based on nearest-neighbor classifier which makes use of a similarity measure, i.e., Euclidean distance to find the training example which is closest to the specified test data. Euclidean distance between two points having coordinates (x, y) and (a, b) is computed by using the equation given below.

$$\text{Dist}((x, y), (a, b)) = \sqrt{(x-a)^2 + (y-b)^2}$$

- NB method is originated from Bayes' theorem with Naïve independent conventions. It predicts the probability of an instance of a dataset belonging to a specific class. The posterior probability of a class *c*, given attribute *x,* is computed by using the equation mentioned below.

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}$$

where $P(x|c)$ represents the probability of attribute x given its class c. $P(x)$ and $P(c)$ are the prior probabilities of attribute x and class c respectively.
J48 [42] is a WEKA classification package of C4.5. It builds the decision tree based on the labeled training data using the concept of entropy.
The main aim of adaptive boosting (AdaBoost) [43] is to transform a group of weak classifiers into a strong classifier. It is a weighted grouping of weak classifiers. It assigns equal weights to all the examples in the dataset. Then the weights of wrongly classified instances are modified. At last, weighted mean of all weak classifiers is considered to take the final decision about the class of an instance.
DT is a type of classifier in which specific features are selected while performing the learning process. It is done by computing cross-validation performance of the table for different sub-sets of attributes and selecting the sub-set which is the best performer. Cross-validation error from a decision table is obtained by manipulating the class counts related to entry of the table. Best first search is usually used to search the feature space.
Bagging (bootstrap aggregation) is a meta-classifier which combines the output of different models to make the final decision more reliable by reducing the variance error [44]. It increases the predictive performance over a single model. A general way to predict the final class is to calculate the weighted average for numeric prediction problems and take a weighted vote for classification problems.
RF is an ensemble method which comprises of various decision trees. It can run effectively on a large dataset. It gives highly accurate results than a single decision tree as it reduces the overfitting [44]. Every decision tree is generated using a random selection of attributes. The class of unknown data is predicted using the aggregation (voting scheme) of predictions made by individual decision trees.
SMO [45] is an optimized version of SVM from WEKA library. It is an optimization algorithm which is used to solve the problem which arises while training SVM. The name of the problem is the quadratic problem. SVM [44] is based on computing a hyperplane with the maximal margin between the data dimensions.

3.3 Feature Selection

A feature selection method is deployed for the selection of top-ranked features which helped in discarding the irrelevant ones. Building a classification model with a lesser number, but a relevant set of features helps in improving its generalization and learning speed. In this work, we have used information gain (IG) [46] method for selecting the top features. It is defined as the extent of information obtained from a feature and is based on the degree of randomness (entropy) in the data. It is calculated by finding the difference between the entropy of data distribution after and before the split.

$$\text{Entropy } = -\sum_i P_i \log_2 P_i$$

where P_i is the class probability.

Ranker algorithm is used for ranking the attributes. Figure 2 shows the top 15 attributes after applying IG + Ranker algorithm. These top rank attributes are used for classification purpose.

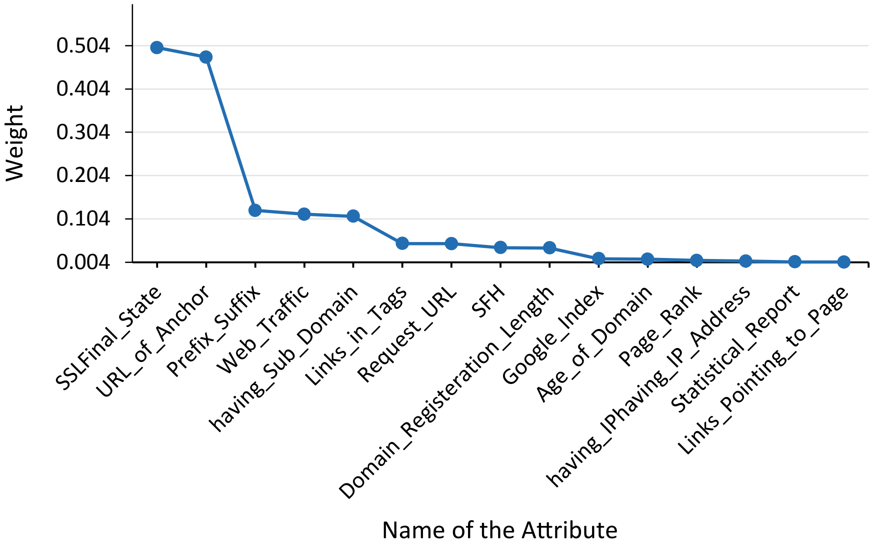

Fig. 2 Top 15 features after feature selection

3.4 Classification after Feature Selection

The selected top 15 features are then employed for building the classification models using the same set of eight algorithms as described earlier. When experiments are performed on both datasets before and after feature selection, the model building time is recorded for carrying out the comparative analysis.

3.5 Evaluation and Validation

A tenfold cross-validation technique has been used for conducting the experiments. This technique works on dividing the original dataset randomly. Firstly, the original dataset is split into ten equal parts. Thereafter, nine parts are used for training and one for testing purpose. The same process is repeated ten times with different combinations. The averaged result is used to measure the performance of algorithms. In order to evaluate machine learning algorithms, we have used various evaluation parameters. These are true positive rate (TPR), false positive rate (FPR), precision, F-measure, accuracy, and Matthews correlation coefficient (MCC) [47]. These are computed by using the fields of confusion matrix as described in Table 1.

- **TPR**: It is also known as recall and is defined as the rate of correctly identified phishing webpages.

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

- **FPR**: It is the rate of incorrectly recognized benign webpages.

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}}$$

- **Precision**: It is a degree of exactness.

Table 1 Confusion matrix

Actual	Classified/predicted as		
	Class	Phishing	Benign
	Phishing	True positive (TP) Phishing webpages predicted correctly as phishing	False negative (FN) Phishing webpages predicted incorrectly as benign
	Benign	False positive (FP) Benign webpages predicted incorrectly as phishing	True negative (TN) Benign webpages predicted correctly as benign

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

- **F-Measure**: It is defined as the harmonic mean of recall and precision.

$$\begin{aligned}\text{F} - \text{Measure} &= \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \\ &= \frac{2 \times \text{TP}}{2 \times \text{TP} + \text{FN} + \text{FP}}\end{aligned}$$

- **Accuracy (%)**: It is the percentage of correctly recognized phishing and benign webpages.

$$\text{Accuracy}(\%) = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{TN} + \text{FP}} \times 100$$

- **MCC**: It is used to evaluate the performance of machine learning algorithms for binary classification. It measures the correlation between the actual and predicted labels and takes the values between −1 and + 1.

$$\text{MCC} = \frac{\text{TP} \times \text{TN} - \text{FP} \times \text{FN}}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}}$$

4 Experimental Results

This section provides a comparative analysis of experimental results along with their pictorial visualizations. All eight classifiers are employed to classify phishing websites by considering the dataset containing 30 features. Table 2 shows the weighted average values of TPR, FPR, precision, F-measure, MCC, and accuracy for

Table 2 Classification results before feature selection (with 30 features)

Classifier	TPR	FPR	Precision	F-measure	MCC	Accuracy (%)
IB1	0.958	0.044	0.958	0.958	0.915	95.78
NB	0.907	0.098	0.907	0.907	0.811	90.7
J48	0.958	0.044	0.958	0.958	0.915	95.82
AdaBoost	0.925	0.077	0.925	0.925	0.848	92.47
Bagging	0.956	0.045	0.956	0.956	0.911	95.59
DT	0.934	0.074	0.935	0.934	0.867	93.41
RF	0.965	0.034	0.965	0.965	0.927	96.52
SMO	0.928	0.077	0.928	0.927	0.853	92.75

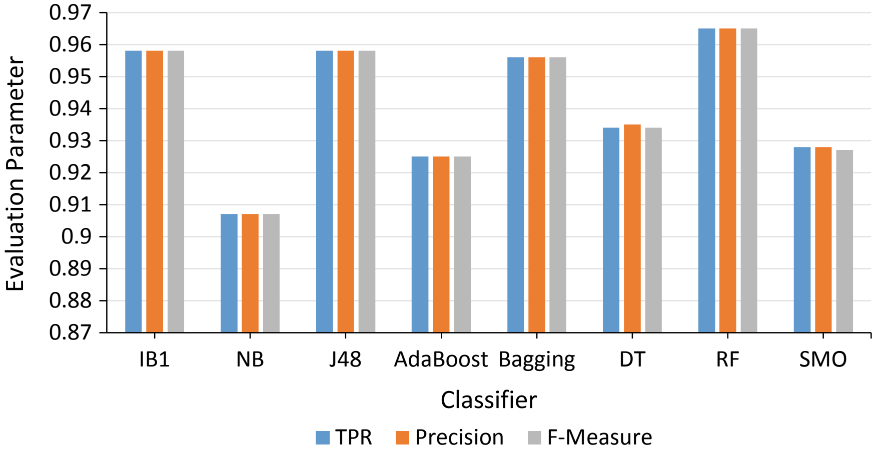

Fig. 3 Comparison of classifiers on the basis of TPR, precision, and F-measure before feature selection

Table 3 Classification results after feature selection (with 15 features)

Classifier	TPR	FPR	Precision	F-measure	MCC	Accuracy (%)
IB1	0.952	0.05	0.952	0.952	0.902	95.16
NB	0.907	0.097	0.907	0.907	0.812	90.73
J48	0.95	0.052	0.95	0.95	0.899	95.03
AdaBoost	0.925	0.077	0.925	0.925	0.848	92.47
Bagging	0.955	0.047	0.955	0.955	0.909	95.5
DT	0.931	0.076	0.932	0.931	0.861	93.13
RF	0.963	0.04	0.963	0.963	0.925	96.3
SMO	0.921	0.087	0.921	0.92	0.839	92.06

the eight machine learning algorithms using 30 features, i.e., before feature selection. It shows that RF provides the highest accuracy of 96.52% followed by J48, IB1, and bagging with 95.82%, 95.78%, and 95.59% accuracy, respectively. RF gives the maximum value of MCC, i.e., 0.928. NB provides the least accuracy of 90.7% with MCC value as 0.811. Figure 3 visualizes the comparison of various classifiers on the basis of TPR, precision, and F-measure before feature selection.

Afterward, the top 15 features are selected using IG feature selection method, and the same set of classification models are again used for the classification purpose. Table 3 shows the weighted average values of TPR, FPR, precision, F-measure, MCC, and accuracy for the eight machine learning algorithms using 15 features selected after employing a feature selection method. It shows that RF provides the highest accuracy of 96.3% followed by bagging, IB1, and J48. Once again, the minimum accuracy is provided by NB, i.e., 90.73%. RF gives the maximum value of MCC,

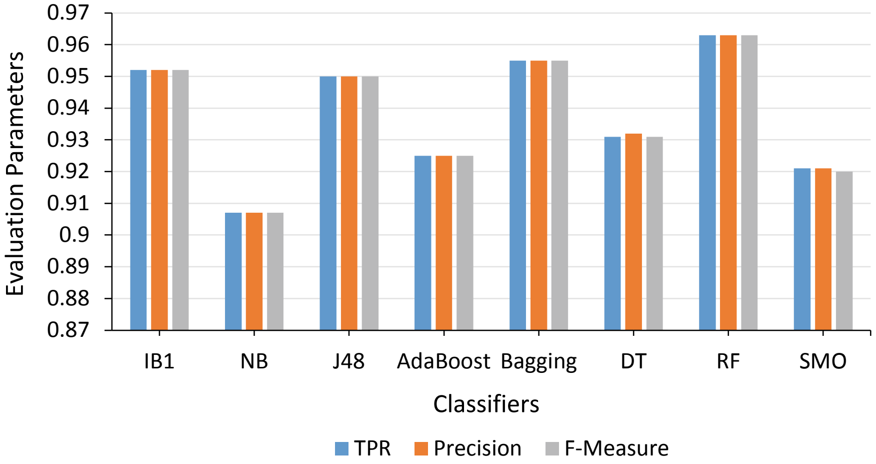

Fig. 4 Comparison of classifiers on the basis of TPR, precision, and F-measure after feature selection

i.e., 0.925. Figure 4 visualizes the comparison of various classifiers on the basis of TPR, precision, and F-measure after feature selection.

Figure 5 depicts the comparison of classifiers on the basis of accuracy before and after features selection. It shows that for almost all the machine learning algorithms considered in this work, the accuracy remains either same or there is insignificant decrement before and after feature selection. It means that the features which are shortlisted by using IG feature selection method are the most relevant ones for the

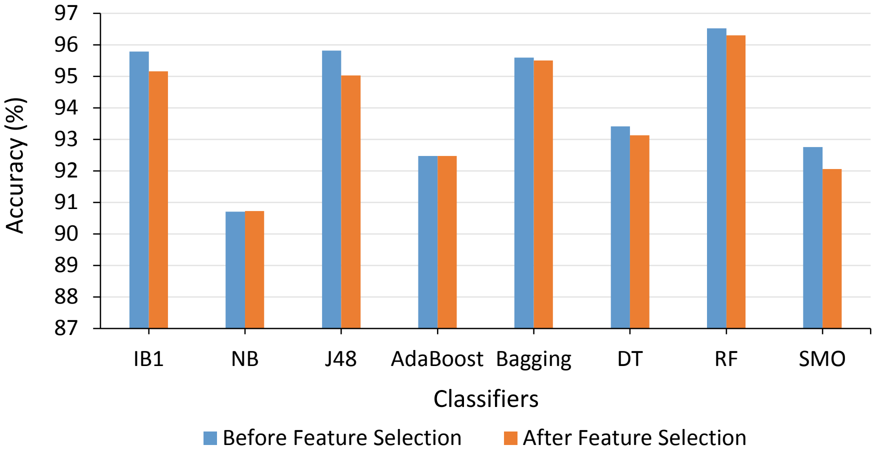

Fig. 5 Comparison of classifiers on the basis of accuracy before and after feature selection

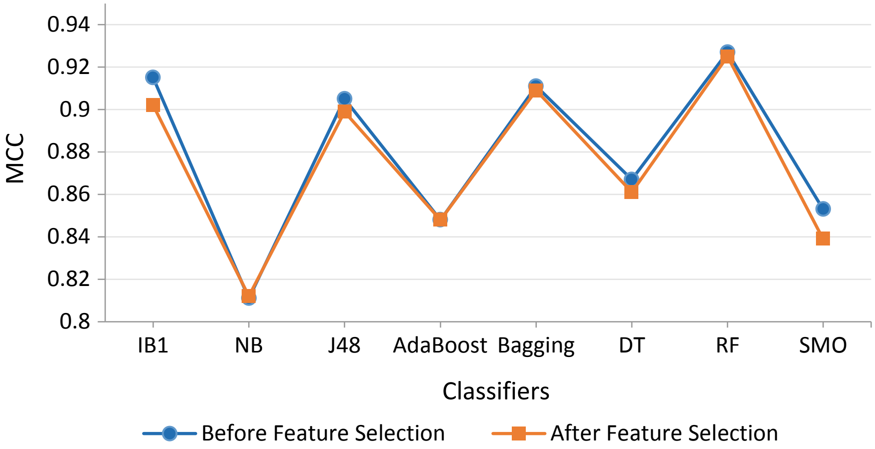

Fig. 6 Comparison of classifiers on the bases of MCC before and after feature selection

Table 4 Model building time (in sec.) before and after feature selection

Classifier	Model building time (in sec.)	
	Before feature selection	After feature selection
IB1	0	0
NB	0.28	0.06
J48	0.62	0.3
AdaBoost	1.14	0.33
Bagging	2.75	1.06
DT	5.39	2.11
RF	5.53	3.43
SMO	19.91	9.27

classification of webpages. A comparison of MCC before and after feature selection is depicted in Fig. 6.

Table 4 shows the build time of classification models while carrying out the classification experiments before and after feature selection. Figure 7 illustrates the comparison of model building time. It shows that for all classifiers, the time taken to build the models is reduced after feature selection. The model build time for IB1 is 0 as it is a lazy learner.

The improvement in model building time does not seem to be as significant since the experiments are conducted using a small dataset. If a large dataset or big data [48, 49] is considered, it would take more time for model building and thus there would be a significant improvement.

In order to detect the new phishing websites, the researchers have used machine learning algorithms. To improve the classification accuracy, they considered a large number of features. However, they do not focus on the problem related to the building

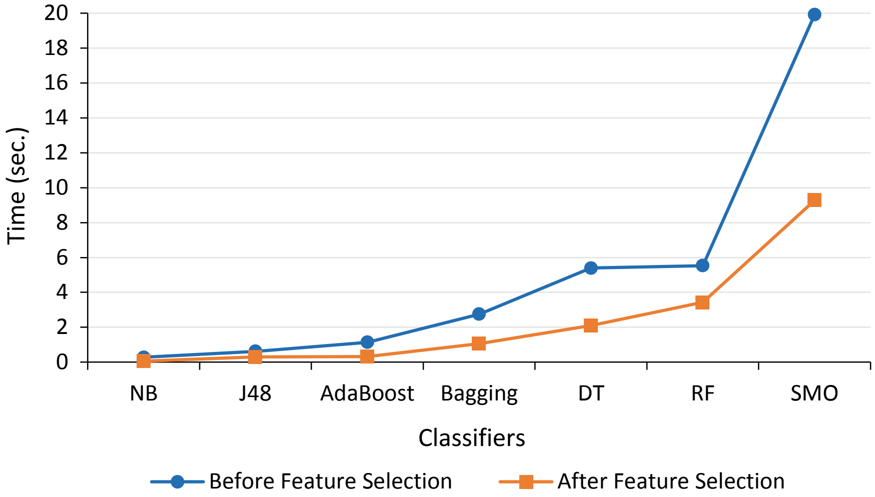

Fig. 7 Comparison of model building time before and after feature selection

time of classification models without compromising the accuracy. In this work, we used a feature selection method to select an appropriate set of features for improving the efficiency of the models without compromising with the accuracy.

5 Conclusion and Future Scope

This chapter presented an efficient system for the detection of phishing websites. It makes use of a feature selection method apart from machine learning algorithms for the purpose of classification. The performance of eight machine learning algorithms is compared before and after feature selection for the classification of webpages into phishing and benign. The experimental results clearly depict that use of a feature selection method for selecting a relevant set of features improves the building time of classification models without a significant reduction in the accuracy for the detection of phishing webpages. Further, the results show that RF gives the best accuracy for both before and after feature selection.

As future work, we intend to use an ensemble of algorithms to compare the accuracy rates with the individual machine learning algorithms. The experimental study can be further extended to understand the performance of deep learning algorithms. We may also enhance our study on large phishing datasets including IoT-based phishing attacks using big data technologies to uncover such attacks more quickly than existing methods.

References

1. Anti-Phishing Working Group (APWG) https://docs.apwg.org//reports/apwg_trends_report_q4_2018.pdf
2. IC3 Annual Report 2018 https://pdf.ic3.gov/2018_IC3Report.pdf
3. Razorthorn phishing report https://www.razorthorn.co.uk/wp-content/uploads/2017/01/Phishing-S
4. Gandotra E, Bansal D, Sofat S (2014) Malware analysis and classification: a survey. J Inf Security 56–65
5. Gupta D, Rani R (2020) Improving malware detection using big data and ensemble learning. Comput Electr Eng 106729
6. Microsoft Security Intelligence Report (2019) vol 24 https://www.microsoft.com/security
7. Logic Bomb Set Off South Korea Cyberattack. https://www.wired.com/2013/03/logic-bomb-south-korea-attack/
8. Los Angeles Times https://www.latimes.com/business/la-fi-mh-anthem-is-warning-consumers-20150306-column.html
9. Threat Analysis Group, Findings on COVID-19 and online security threats https://blog.google/technology/safety-security/threat-analysis-group/findings-covid-19-and-online-security-threats/
10. Selenium https://docs.seleniumhq.org/download/
11. Gandotra E, Bansal D, Sofat S (2016) Tools and techniques for malware analysis and classification. Int J Next-Gener Comput
12. Jsoup Java HTML Parser, with best of DOM, CSS, and jquery https://jsoup.org/
13. OpenDNS, PhishTank https://wwwphishtank.com/
14. Google Safe Browsing API—Google Code https://code.google.com/apis/safebrowsing/
15. Seifert C, Welch I, Komisarczuk P (2008) Identification of malicious web pages with static heuristics. In: 2008 Australasian Telecommunication Networks and Applications Conference, IEEE, pp 91–96
16. Jain AK, Gupta BB (2017) Phishing detection: analysis of visual similarity based approaches. Secur Commun Network
17. Gandotra E, Bansal D, Sofat S (2015) Computational techniques for predicting cyber threats. In: Intelligent computing, communication and devices, pp 247–253, Springer, New Delhi
18. Tan CL, Chiew KL, Wong K (2016) PhishWHO: phishing webpage detection via identity keywords extraction and target domain name finder. Decision Support Systems, pp 18–27
19. Chiew KL, Chang EH, Tiong WK (2015) Utilisation of website logo for phishing detection. Comput Security 16–26
20. Jain AK, Gupta BB (2018) Towards detection of phishing websites on client-side using machine learning based approach. Telecommun Syst 687–700
21. Srinivasa Rao R, Pais AR (2017) Detecting phishing websites using automation of human behavior. In: Proceedings of the 3rd ACM workshop on cyber-physical system security, ACM, pp 33–42
22. Sahingoz OK, Buber E, Demir O, Diri B (2019) Machine learning based phishing detection from URLs. Expert Syst Appl 345–357
23. Gu X, Wang H, Ni T (2013) An efficient approach to detecting phishing web. J Comput Inf Syst 5553–5560
24. Moghimi M, Varjani AY (2016) New rule-based phishing detection method. Expert systems with applications, pp 231–242
25. Xiang G, Hong J, Rose CP, Cranor L (2011) Cantina+ a feature-rich machine learning framework for detecting phishing web sites. ACM Transactions on Information and System Security (TISSEC), pp 1–28
26. Zhang Y, Hong JI, Cranor LF (2007) Cantina: a content-based approach to detecting phishing web sites. In: Proceedings of the 16th international conference on World Wide Web, ACM, (2007) pp 639–648

27. Joshi A, Pattanshetti P, Tanuja R (2019) Phishing Attack Detection using Feature Selection Techniques. In: Nutan College of Engineering & Research, International Conference on Communication and Information Processing (ICCIP)
28. Wu CY, Kuo CC, Yang CS (2019) A phishing detection system based on machine learning. In: 2019 International Conference on Intelligent Computing and its Emerging Applications (ICEA), pp 28–32
29. Zamir A, Khan HU, Iqbal T, Yousaf N, Aslam F, Anjum A, Hamdani M (2020) Phishing web site detection using diverse machine learning algorithms. The Electronic Library
30. Almseidin M, Zuraiq AA, Al-kasassbeh M, Alnidami N (2019) Phishing detection based on machine learning and feature selection methods. Int J Interactive Mobile Technol (iJIM) 171–183
31. Yerima SY, Alzaylaee MK (2020) High accuracy phishing detection based on convolutional neural networks. arXiv preprint arXiv:2004.03960
32. Basnet RB, Doleck T (2015) Towards developing a tool to detect phishing URLs: a machine learning approach. In 2015 IEEE International Conference on Computational Intelligence & Communication Technology, IEEE, pp 220–223
33. Hurrah NN, Parah SA, Loan NA, Sheikh JA, Elhoseny M, Muhammad K (2019) Dual watermarking framework for privacy protection and content authentication of multimedia. Future Gener Comput Syst 654–673
34. Parah SA, Sheikh JA, Bhat GM (2014) Fragility evaluation of intermediate significant bit embedding (ISBE) based digital image watermarking scheme for content authentication. In: 2014 International conference on advances in electronics computers and communications, IEEE pp 1–6
35. Gull S, Loan NA, Parah SA, Sheikh JA, Bhat GM (2018) An efficient watermarking technique for tamper detection and localization of medical images. J Ambient Intell Humanized Comput pp 1–10
36. Gull S, Parah SA (2019) Color image authentication using dual watermarks. In: 2019 fifth international conference on image information processing (ICIIP), pp 240–245
37. Giri KJ, Bashir R, Bhat JI (2019) A discrete wavelet based watermarking scheme for authentication of medical images. Int J E-Health Med Commun (IJEHMC), pp 30–38
38. UCI Machine Learning Repository, “Phishing Websites Dataset” https://archive.ics.uci.edu/ml/datasets/phishing+websites
39. Mohammad RM, Thabtah F, McCluskey L (2012) An assessment of features related to phishing websites using an automated technique. In 2012 International conference for internet technology and secured transactions, IEEE pp 492–497, IEEE
40. Alexa Most Popular sites. https://www.alexa.com/topsites
41. Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH (2009) The WEKA data mining software: an update. ACM SIGKDD explorations newsletter, pp 10–18
42. Quinlan JR (2014) C4.5: Programs for Machine Learning. Elsevier
43. Schapire RE (1990) The strength of weak learnability. Machine Learning, pp 197–227
44. Witten IH, Frank E (2002) Data mining: practical machine learning tools and techniques with Java implementations. Acm Sigmod Record pp 76–77
45. Platt J (1998) Sequential minimal optimization: a fast algorithm for training support vector machines
46. Gandotra E, Bansal D, Sofat S (2016) Zero-day malware detection. In: 2016 sixth international symposium on embedded computing and system design (ISED), IEEE, pp 171–175
47. Gandotra E, Bansal D, Sofat S (2017) Malware threat assessment using fuzzy logic paradigm. Cybern Syst 29–48
48. Gupta D, Rani R (2019) A study of big data evolution and research challenges. J Inf Sci 322–340 (2019)
49. Gupta D, Rani R (2018) Big data framework for zero-day malware detection. Cybern Syst 103–121