Contents lists available at GrowingScience

International Journal of Data and Network Science

homepage: www.GrowingScience.com/ijds

A new phishing-website detection framework using ensemble classification and clustering

Mohammad A. Alsharaiah[a], Ahmad Adel Abu-Shareha[a*], Mosleh Abualhaj[b], Laith H. Baniata[c], Omar Adwan[d,e], Adeeb Al-saaidah[b] and Majdi Oraiqat[f]

[a]*Department of Data Science and Artificial Intelligence, Al-Ahliyya Amman University, Amman, Jordan*
[b]*Department of Networks and Information Security, Al-Ahliyya Amman University, Amman, Jordan*
[c]*Gachon University, South Korea*
[d]*Department of Software Engineering, Al-Ahliyya Amman University, Amman, Jordan*
[e]*The University of Jordan, Amman, Jordan*
[f]*Engineering and Artificial Intelligence Department, Al-Balqa Applied University, Jordan*

C H R O N I C L E

Article history:
Received: October 20, 2022
Received in revised format: October 28, 2022
Accepted: January 7, 2023
Available online: January 7, 2023

Keywords:
Ensemble Learning
Classification
Clustering
Phishing Detection

A B S T R A C T

Phishing websites are characterized by distinguished visual, address, domain, and embedded features, which identify and defend such threats. Yet, phishing website detection is challenged by overlapping these features with legitimate websites' features. As the inter-class variance between legitimate and phishing websites becomes low, commonly utilized machine learning algorithms suffer from low performance in overlapping feature cases. Alternatively, ensemble learning that combines multiple predictions intending to address low inter-class variations in the classified data improves the performance in such cases. Ensemble learning utilizes multiple classifiers of similar or different types with multiple deviations of the training data. This paper develops a framework based on random forest ensemble techniques. The limitations of the random forest are the inability to capture the high correlation between features and their join dependency on the label. The random forest is combined with k-means clustering to capture the feature correlation. The framework is evaluated for phishing detection with a dataset of 5000 samples. The results showed the proposed framework over-performed the random forest classifier, all other ensemble classifiers, and the conventional classification algorithms. The proposed framework achieved an accuracy of 98.64%, precision of 0.986, recall of 0.987, and F-measure of 0.986.

1. Introduction

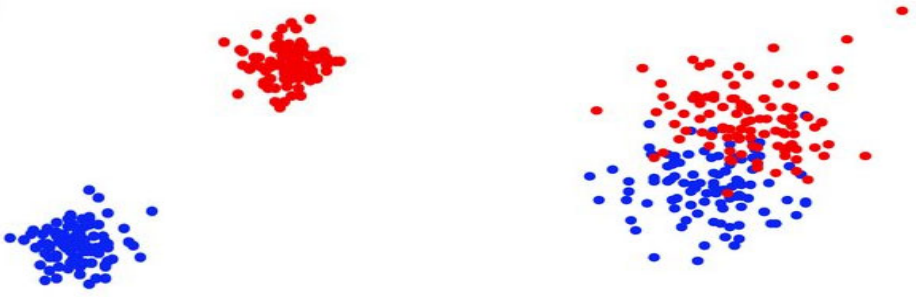
A malicious phishing website is developed by mimicking a legitimate website to deceive its users. The phisher aims at thieving the information provided by the user as he/she is convinced to be using the legitimate website of a trusted party. The threats of phishing websites lie in the targeted information, such as login details, financial information, business-related information, etc. The threats of phishing websites increase as the number of phishing websites increases rapidly, mainly targeting important information of unaware users (Ganesan, 2022). The Anti-Phishing Working Group (APWG) observed 1,097,811 phishing attacks during the second quarter of 2022, a new record for the phishing attacks observed by the APWG in any quarter. Social media threats continuously increase, reaching 47% percent of total attacks in the same quarter. The report also showed increased mobile-phone threats and money loss (APWG, 2022). Besides, Microsoft Security Intelligence's first-half report of 2022 indicated that "credential phishing schemes are on the rise and are a substantial threat to users everywhere because they indiscriminately target all inboxes" to deceive users of the phishing websites (Intelligence, 2022).

* Corresponding author.
E-mail address: a.abushareha@ammanu.edu.jo (A. A. Abu-Shareha)

ISSN 2561-8156 (Online) - ISSN 2561-8148 (Print)

doi: 10.5267/j.ijdns.2023.1.003

Detecting phishing websites depends on features that characterize and distinguish these websites from legitimate ones, such as the visual, address, and domain features. The visual appearance is one of the core components that the attacker relies on for deceiving the users, as the phishing website is developed with the same visual interface as the legitimate one. Yet, the underlying content of the website embodied in the HTML features could be significantly different from the content of legitimate websites. Besides, the domain living period can be efficiently relied on for identifying phishing websites, which commonly have short lives (Aljofey et al., 2022). Moreover, the length of the address and its content are also used for detection, as it contains special characteristics compared to the address of legitimate websites. However, the inter-class variation (see Fig. 1) of the website classes (i.e., phishing and legitimate) is low, which resulted in the inability to distinguish the phishing website from legitimate ones (Akpan & Starkey, 2021).



a) High Inter-Class Variance b) Low Inter-Class Variance

Fig. 1. Inter-Class Variance Example

Data classification algorithms are used to detect phishing websites, which are implemented in two main phases, training and testing. Training the classifiers is achieved by samples of legitimate and phishing classes. The trained model is then used to classify unknown websites into legitimate or phishing classes. The preprocessing steps, feature normalization, and feature selection are commonly used before the classification process to ease and improve the output results. Various classification algorithms were used for the phishing detection problem to improve the results' accuracy and reduce the predicting processes' complexity. The classified dataset is formed by features that affect the utilized classifiers' performance. These features commonly include visual, address, domain, and embedded features, which greatly overlap phishing and legitimate websites. Commonly utilized machine learning algorithms suffer from low performance in overlapping feature cases as the inter-class variations become low.

The overlapping between features can be described using two terms, the inter-class and intra-class variations, as illustrated in Fig. 1. The intra-class variation is the compactness of the samples of a single class in the feature space. The effect of the intra-class low variance is related to the ability to learn a good model for that class. The classifier learns a limited variance of the class, which results in bad performance when the testing data belongs to a higher variance group. The variability of the legitimated websites to be mimicked and the variability of the corresponding phishing will create high intra-class variability in the feature space. The problem lies in the inter-class variance, the frontier between the samples of the different classes in the feature space. The similarity between legitimate and phishing websites from various aspects, especially in the visual features, creates low inter-class variance in the feature space. The low inter-class variation affects the performance of the classification algorithms significantly.

The ensemble learning techniques combine multiple predictions to address the classified data variations to improve the performance in such cases. Ensemble learning utilizes multiple classifiers of similar or different types with multiple deviations of the training data. Thus, each prediction can focus on different aspects, which eases the low inter-class variance problems. Random forest is an ensemble classifier known for its accurate results depending on combinations of multiple-weak classifiers. The limitations of the random forest are the inability to capture the high correlation between features and their joint dependency on the label. In this paper, a framework for phishing detection based on ensemble learning is developed and evaluated. The random forest is combined with k-means clustering to capture the feature correlation and improve the performance of phishing detection. The rest of this paper is organized as follows. Section 2 presents the related work on phishing detection. Section 3 discusses the details of the proposed framework and its components. Section 4 presents the experiments and the results. Finally, the conclusion of this paper is given in Section 5.

2. Related Work

Generally, phishing detection techniques can be classified into blacklist-based, similarity-based, and prediction-based. The blacklist-based applications, such as Netcraft extension, maintains an updated list of the URL of the phishing websites to alert or prevent the user from browsing these websites. Yet, this technique is very poor at defending the new phishing websites developed daily. Accordingly, even when the URL is captured, the attacker can easily move the phishing website to another

URL (Li & Helenius, 2007). The similarity-based, such as the Spoofguard (Krishnan & Subramaniyaswamy, 2015), implements multiple-level validation by checking the domain with the recently accessed domains for any possible tweaking that the user may not recognize. Moreover, the URL is checked for suspicious embedded usernames, invalid port numbers, etc. The problem with these approaches is their limited capabilities in capturing continuously created phishing websites.

The prediction-based can be content-based or URL-based, or a combination of them depending on the utilized and extracted features. The prediction-based used supervised and unsupervised machine learning techniques to classify a website as phishing or legitimate. The supervised machine learning technique is preferred as the results produced the final label. At the same time, the unsupervised might need an extra processing step, as it only groups samples in multiple groups, regardless of the label provided (Rendall, Nisioti, & Mylonas, 2020). Generally, the classification algorithms can be classified into five groups, instance-based, probability-based, artificial neural network (ANN), support vector machine (SVM), and decision tree (DT)-based classifications. Each of these groups has advantages and disadvantages.

The instance-based does not build any training model; instead, the training samples are used in the predicting phase. The instance-based is simple to implement. The disadvantages of the instance-based model are time and memory expenses during the prediction phase, sensitivity to outliers and noise, and biased toward the class of majority samples (He, Sheng, Liu, & Zou, 2021). The probability-based built probability model is based on the posterior of the features of each class. The constructed model is robust to noise yet, does not perform well with complex cases with high features correlations. The artificial neural network trains a network using the training samples, which commonly results in accurate results in complex cases with outliers, noise, and incomplete information. The disadvantages of the network are the instability and the inability to choose the optimal network structure. The decision tree builds a training model in the form of a tree that is simple and easy to implement and analyze. Yet, the disadvantages of the decision tree are noise effects and the inability to address complex cases. The SVM is well known for its performance, which built a hyperplane between two data classes. The complexity of this model, especially in multi-class classification problems, and the noise effects are the main disadvantage of this classifier (Sen, Hajra, & Ghosh, 2020).

PhishAri was developed as a browser extension by Aggarwal, Rajadesingan, and Kumaraguru (2012) for detecting phishing URLs embodied in tweets using the Random Forest (RF) classification technique. PhishAri is based on URL features only and achieved an accuracy of 92.52%. Various supervised machine learning algorithms, such as RF, were used with an accuracy of 97.36%, as reported by Subasi, Molah, Almkallawi, and Chaudhery (2017), and 96.17%, as reported by Chiew et al. (2019). Convolutional Neural Network (CNN) was also used by (Y. Li, Yang, Chen, Yuan, & Liu, 2019), which achieved an accuracy of 98.60%.

Ensemble learning techniques were used to address the inter-class variation and improve the output results. One of the commonly utilized ensemble techniques is voting. The core functionality of the voting ensemble technique is using several classifiers, each of which is trained to recognize the underlying classes. In the testing phase, every trained classifier initiates a prediction, and the majority voting determines the overall prediction. Bootstrap Aggregating or bagging is used to decrease the variance and aids in eluding overfitting issues. Bagging constructs several identical classifiers on a small portion of the population. The Bagging method depends on the model averaging voting techniques, as shown in Fig. 2. It is typically applied to the random forest when it combines several random decision trees.

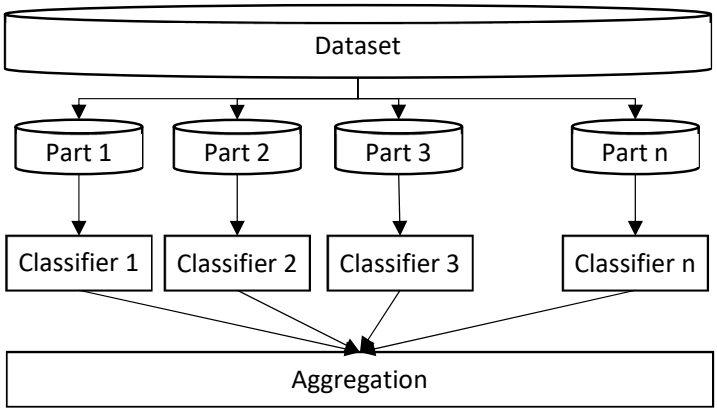


Fig. 2. Bagging Ensemble Classification

Adaboost is a statistical classification algorithm with other methods (Freund & Schapire, 1997). In the prediction phase, the results from utilized weak learners are combined using a weighted sum. Besides, Adaboost can run on top of strong base learners like decision trees, which leads to a more precise model (Hastie, Rosset, Zhu, & Zou, 2009). The Adaboost is less susceptible to the machine learning issues such as overfitting ("Explaining the Success of AdaBoost and Random Forests as Interpolating Classifiers," 2017).

In the literature, Sahingoz, Buber, Demir, and Diri (2019) implemented DT, k-star, AdaBoost, k-nearest neighborhood (KNN), self-optimization map (SOM), RF, and Naïve Bayes (NB) classification methods based on URL features for phishing detection. The developed methods achieved an accuracy of 97.98%. Alsariera, Elijah, and Balogun (2020) developed phishing detection using boosting and bagging. Based on the extra-tree weak classifier, based on four-ensemble learning, AdaBoost, Bagging, Rotation Forest, and LogitBoost. The developed method achieved an accuracy of 97%. Subasi and Kremic (2020) developed AdaBoost and Multiboost ensemble learning for phishing detection, which improved the detection performance and achieved an accuracy of 97.61%. The results were evaluated using F-measure and ROC area. Y. Li et al. (2019) developed Gradient boosted decision tree, light gradient boosting machine (LightGBM), and XGradientBoost using HTML and URL features and achieved an accuracy of 97.3%.

Generally, various approaches were developed for phishing detection using traditional machine learning and ensemble techniques, which suffer from low performance. The ensemble techniques provide a better performance, each with its limitations. The RF, based on weak decision tree classifiers, suffers from joint correlation with the class label. Accordingly, RF is integrated with K-Means clustering, which captures the feature dependencies.

3. Proposed Work

A framework for phishing detection is proposed to improve the accuracy of phishing detection, as illustrated in Fig. 3.

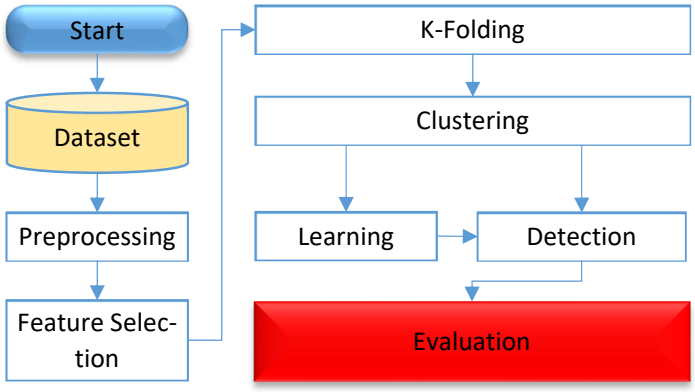


Fig. 3. The Proposed Phishing Detection Framework

3.1 Data Preprocessing

Data transformation and normalization are implemented on the input data to eliminate the influence of outliers and features with wide and inconsistent ranges and ease the classification step. In the transformation step, the nominal and ordinal data are transformed into numerical data using the OneHot technique. The OneHot technique creates multiple columns for each column in the dataset. The number of the created columns depends on the length of the value set of each column, as illustrated in Fig. 4. In the normalization step, both min-max normalization and z-score based are implemented. As given in Eq. (1), the min-max scalar converted the values of a specific attribute into the range [0-1]. On the other hand, for the z-score scaler, as given in Eq. (2), the output value will be in the range of [-3std – 3std].

Protocol
TCP
UDP
ICMP

Protocol-TCP	Protocol-UDP	Protocol-ICMP
1	0	0
0	1	0
0	0	1

Fig. 4. OneHot Encoder

$$z = (x - min)/(\max - min) \quad (1)$$

$$z = (x - \mu)/\sigma \quad (2)$$

3.2 Feature Selection

In the feature selection step, wrapper- and filter-based methods are implemented to extract the most significant subsets of features. The wrapper-based used the classification output to select the optimal subset of features collectively. Accordingly, for its adequate performance and ease of approach, an instance-based classifier is used as the base classifier for the forward algorithm for wrapper-based selection. The information gain (IG) is filter-based, in which the evaluation of the features is

implemented regardless of the classification output. Moreover, Principle component analysis (PCA) is used as the third method for feature selection.

3.3 Clustering

The k-Means clustering algorithm is implemented to create 2-clusters of samples. Each sample's cluster number is added as the dataset's new attribute/column. Accordingly, the added feature captures the association between the features.

3.4 Classification

In the final step, the classification algorithms are implemented to classify the input samples and detect phishing. Random Forest (RF) used randomly selected features to construct sibling decision trees in the training phase. In the testing step, voting over the output of the different trees is implemented to find the final class of the input sample, as illustrated in Fig. 5 (HO, 1995; Ziegel, 2003).

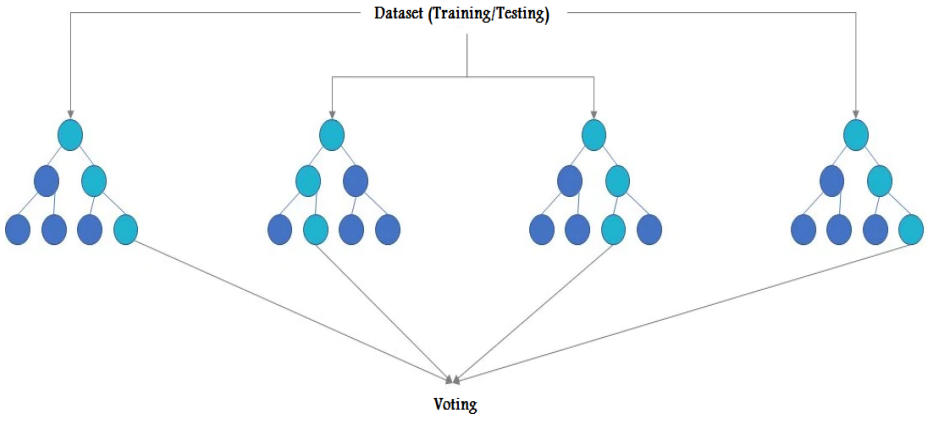


Fig. 5. Random Forest Functionality for Classification Task

4. Experiments and Results

The experiments are conducted using Python programming language and Pandas, Numpy, and Scikit libraries. The dataset, evaluation metrics, and results are discussed in this section. The performance of the proposed framework is compared to the simple machine-learning techniques and various ensemble machine-learning algorithms, including bagging, AdaBoost, voting, and RF algorithms implemented.

4.1 Phishing Dataset

A phishing dataset utilized in the experiments is described by Chiew et al. (2019). The dataset consists of 48 features and 10,000 samples, 5,000 phishing websites, and 5,000 legitimate ones. The phishing samples were collected from the OpenPhish and Phish Tank, while the legitimate samples were gathered from different resources such as Common Crawl and Alexa. These samples were collected from January to May 2015 and May to June 2017. The features of this dataset can be classified into three classes: HTML/JavaScript features, abnormal features, and address bar features. The HTML/JavaScript depends on the tags and fragments in the source code of the collected websites. The address bar features depend on the port number and the URL length. The abnormal features are the actions performed on the website, like downloading things from outside the domain (Zabihimayvan & Doran, 2019).

4.2 Evaluation Measures

The proposed framework is evaluated using the accuracy and the confusion matrix metrics (Tharwat, 2020). Besides, precision, recall, and F-score measures are also used for the evaluation. The confusion matrix (i.e., the error matrix) is a statistical classification that visualizes the model's performance, as shown in Fig. 6.

		Predicted	
		Negative	Positive
Actual	Negative	True Negative Prediction	False Positive Prediction
	Positive	False Negative Prediction	True Positive Prediction

Fig. 6. General Representation of the Confusion Matrix (For Binary Classification)

The confusion matrix exposes the various indicators. The True Positive (TP) represents the true prediction counts of the positive data points, which counts the predicted value as positive for samples of actual values likewise positive. The False Positive (FP) represents the counts of the negative value falsely classified as positive. The True Negative (TN) represents the number of truly predicted negative data points in which the predicted value is negative, and the actual value is negative. Finally, False Negative (FN) represents the number of positive values wrongly classified as negative.

The accuracy metric is the ratio of the correctly classified instances, which used the TN and TP indicators, as given in Eq. (3). The precision is computed as the ratio of TP divided by the number of positively labeled samples by the utilized classifier, as given in Eq. (4). The recall measure is calculated as the number of TP divided by the number of positive samples in the dataset, as given in Eq. (5). Finally, the F-score calculated in Eq. (6) is used for averaging the precision and recall measures. The output results of these measures are on the scale of [0-1] or can be represented as a percentage of 100%.

$$Accuracy = TP + TN \,/\, (TP \; + TN \; + FP + FN) \tag{3}$$
$$Precision = TP \,/\, (TP \; + FP) \tag{4}$$
$$Recall = TP \,/\, (TP \; + FN) \tag{5}$$
$$F - Measure = 2 * Precision * Recall \,/\, (Precision \; + Recall\,) \tag{6}$$

4.3 Results

In the experiments, a 10-fold cross-validation method is employed to minimize the estimation variance, in which the dataset is divided into 10 folds (subsets). Every fold is used as a testing fold, while the rest are used to build the model in the training phase. In the execution of the initial experiment, parameters adjustment is implemented. Next, the experiments are conducted to report the performance of the detection processes.

The results, as reported in Table 1 and illustrated in Fig. 7, can be summarized as follows, the proposed framework achieved the best results with an accuracy of 98.64%. AdaBoost has achieved an accuracy of 94.23%, while Bagging achieved an accuracy of 97.33%. The voting model with the combination of the J48 classifier achieved better results compared to Ada-boost. Still, the voting model results are less than the Bagging model in terms of accuracy. The overall accuracy for the voting model is 96.25%. The random forest achieved the best performance of 98.37%. Accordingly, the proposed framework achieved the best results compared to the other classification algorithms. Moreover, the proposed method has comparable results with the results reported in the literature.

Table 1
The Performance Results of the Ensemble Techniques

Technique	TP	FP	Precision	Recall	F-measure	Accuracy
KNN	0.955	0.045	0.955	0.955	0.955	95.53%
NB	0.852	0.14.9	0.864	0.852	0.850	85.15%
SVM	0.939	0.061	0.939	0.939	0.939	93.87%
DT	0.960	0.040	0.960	0.960	0.960	95.98%
ANN	0.966	0.034	0.966	0.966	0.966	96.59%
AdaBoost	0.940	0.058	0.942	0.942	0.942	94.23%
Bagging	0.973	0.027	0.973	0.973	0.974	97.33%
Voting (J48)	0.963	0.038	0.963	0.963	0.962	96.25%
Random Forest	0.984	0.016	0.984	0.984	0.984	98.37%
Proposed Framework	**0.987**	**0.013**	**0.986**	**0.987**	**0.986**	**98.64%**

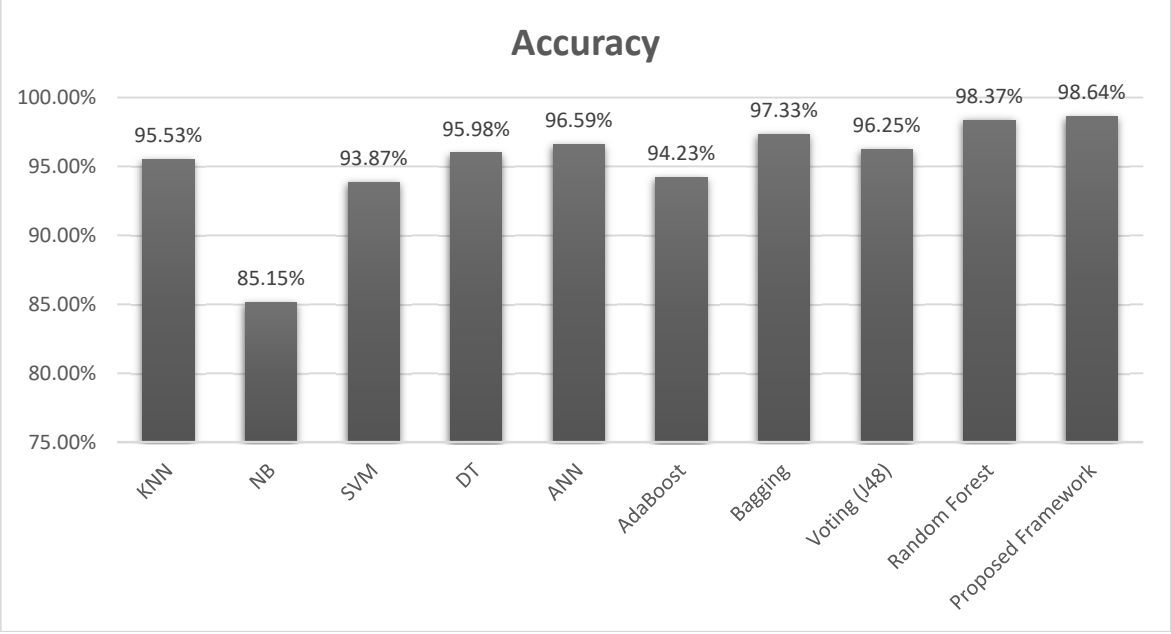


Fig. 7. The Accuracy of the Ensemble Techniques

5. Conclusion

This paper implements a framework for phishing website detection based on an effective machine learning technique of pre-processing, feature selection, and classification algorithms and techniques. The ensemble-based random forest technique achieved the best results among the compared methods. The results showed that all ensemble techniques over-performed the classical classification algorithms. Moreover, it is also noted that feature selection, which is commonly ignored while using ensemble techniques, is useful in improving performance. Moreover, feature selection using different techniques lead to different results. The same rules are applied to the preprocessing steps. Among the utilized ensemble learning techniques, Random Forest achieved the best performance with an accuracy of 98.64%. Accordingly, the results suggest that using multiple classifiers improves phishing detection instead of using a single classifier with low performance with this type of data.

References

APWG, T. A.-P. W. G. (2022). Phishing Activity Trends Reports. Retrieved from https://apwg.org/trendsreports/

Aggarwal, A., Rajadesingan, A., & Kumaraguru, P. (2012). *PhishAri: Automatic realtime phishing detection on twitter.* Paper presented at the 2012 eCrime Researchers Summit.

Akpan, U. I., & Starkey, A. (2021). Review of classification algorithms with changing inter-class distances. *Machine Learning with Applications, 4*, 100031.

Aljofey, A., Jiang, Q., Rasool, A., Chen, H., Liu, W., Qu, Q., & Wang, Y. (2022). An effective detection approach for phishing websites using URL and HTML features. *Scientific Reports, 12*(1), 1-19.

Alsariera, Y. A., Elijah, A. V., & Balogun, A. O. (2020). Phishing website detection: forest by penalizing attributes algorithm and its enhanced variations. *Arabian Journal for Science and Engineering, 45*(12), 10459-10470.

Chiew, K. L., Tan, C. L., Wong, K., Yong, K. S., & Tiong, W. K. (2019). A new hybrid ensemble feature selection framework for machine learning-based phishing detection system. *Information Sciences, 484*, 153-166.

Freund, Y., & Schapire, R. E. (1997). A desicion-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences, 55*(1), 119-139.

Ganesan, S. (2022). Detection of Phishing Websites Using Classification Algorithms. In *Cyber Security and Digital Forensics* (pp. 129-141): Springer.

Hastie, T., Rosset, S., Zhu, J., & Zou, H. (2009). Multi-class adaboost. *Statistics and Its Interface, 2*(3), 349-360.

He, Z., Sheng, C., Liu, Y., & Zou, Q. (2021). Instance-based classification through hypothesis testing. *IEEE Access, 9*, 17485-17494.

HO, T. K. (1995). *Random Decision Forests*. Paper presented at the Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC, Canada

Intelligence, M. S. (2022). Microsoft Digital Defense Report Retrieved from https://www.microsoft.com/en-us/security/business/microsoft-digital-defense-report-2022

Krishnan, D., & Subramaniyaswamy, V. (2015). Phishing website detection system based on enhanced itree classifier. *ARPN Journal of Engineering Application Science, 10*(14), 5688-5699.

Li, L., & Helenius, M. (2007). Usability evaluation of anti-phishing toolbars. *Journal in Computer Virology, 3*(2), 163-184.

Li, Y., Yang, Z., Chen, X., Yuan, H., & Liu, W. (2019). A stacking model using URL and HTML features for phishing webpage detection. *Future Generation Computer Systems, 94*, 27-39.

Rendall, K., Nisioti, A., & Mylonas, A. (2020). Towards a multi-layered phishing detection. *Sensors, 20*(16), 4540.

Sahingoz, O. K., Buber, E., Demir, O., & Diri, B. (2019). Machine learning based phishing detection from URLs. *Expert Systems with Applications, 117*, 345-357.

Sen, P. C., Hajra, M., & Ghosh, M. (2020). Supervised classification algorithms in machine learning: A survey and review. In *Emerging technology in modelling and graphics* (pp. 99-111): Springer.

Subasi, A., & Kremic, E. (2020). Comparison of adaboost with multiboosting for phishing website detection. *Procedia Computer Science, 168*, 272-278.

Subasi, A., Molah, E., Almkallawi, F., & Chaudhery, T. J. (2017). *Intelligent phishing website detection using random forest classifier.* Paper presented at the 2017 International conference on electrical and computing technologies and applications (ICECTA).

Tharwat, A. (2020). Classification assessment methods. *Applied Computing and Informatics*.

Zabihimayvan, M., & Doran, D. (2019). *Fuzzy rough set feature selection to enhance phishing attack detection.* Paper presented at the 2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE).

Ziegel, E. R. (2003). The elements of statistical learning. In: Taylor & Francis.