Comparative Analysis of Ensemble Learning and Non-Ensemble Machine Learning Algorithms for Phishing URL Detection

[1]Chiamaka M. Igwilo and *[2]Victor T. Odumuyiwa
[1]Department of Computer Science, African University of Science and Technology, Abuja, Nigeria
[2]Department of Computer Science, University of Lagos, Lagos, Nigeria
cigwilo@aust.edu.ng|vodumuyiwa@unilag.edu.ng

Received: 22-FEB-2022; Reviewed: 25-MAR-2022; Accepted: 10-SEP-2022
https://doi.org/10.46792/fuoyejet.v7i3.807

ORIGINAL RESEARCH

Abstract- Phishing is a social engineering attack that has been perpetuated for long and is still a prominent attack with an attending high number of victims. Through phishing, attackers can gain easy access to sensitive information about a company or an individual. This research compares the import of features such as lexical features, Domain Named Based features, HTML Features, and tokenization of URLs in detecting phishing URLs. Experimental procedures were designed to compare the efficiency of the four categories of features used separately on three machine learning models (K-Nearest Neighbour, Decision Tree, Logistic Regression) and five ensemble learning classifiers (Random Forest, Bagging, Stacking, Ada Boost, Gradient Boost). Results obtained show higher accuracy for experiments done using URL tokenization with stacking classifier with accuracy scores of 96% and 99.3% respectively for the two datasets used. Future study would be based on more dataset with larger sample size to provide a basis for generalisation.

Keywords- Classification, Ensemble Learning, Phishing detection, URL features

1 INTRODUCTION

With the advancement of technology over the years and the tremendous growth in data generation through activities on social networks, the Internet as well as Internet of Things (IoT) devices; the need for data privacy, protection, and security against cyber-attacks cannot be over-emphasized (Sikdar, 2020). While attackers keep developing new ways to gain unauthorized access to networks, programs, and data, phishing still remains one of the oldest and most prominent methods they use. Due to the COVID-19 pandemic, a large amount of workload and business-related projects are being carried out over the Internet from home. Cybercriminals are upgrading tactics and exploring the technological challenges faced in securing data while working away from the offices. Working from home has become an avenue for increasing data theft, fraudulent emails, spam, and phishing attempts. In addition, as more services and objects become electronically stored, the number of internet users will increase and cyber-attacks will increase as well (Odumuyiwa & Analogbei, 2021).

According to reports and the article published by Kuala Lumpur in Deloitte, it states that "91% of all cyber-attacks begin with a phishing email to an unexpected victim and 32% of all successful breaches involve the use of phishing techniques" (Lumpur, 2020). Despite the knowledge of several phishing attacks over the years, individuals are still falling victims to this oldest form of cyberattack. In Nigeria, the issue of phishing and cybercrimes are still ravaging the economic sector of the country. Adepetun (2019) reports that Nigeria continues to lose over 127 billion Naira yearly due to Nigeria's exposure to phishing attacks. The report published by Ogbonnaya shows that in 2018, Nigeria's commercial banks lost a total of $39 million (15 billion Naira) to cybercrimes and electronic fraud (Ogbonnaya, 2020). His report illustrates that majority of these crimes were done through phishing and identity theft, contributing to an increase over the previous year's loss of $7.1 million (2.37 billion Naira) to the same crimes.

Phishing is a form of fraud whereby an attacker tries to access sensitive information such as account and login details by sending an email to a person disguising the source of the email as though it is from a reliable organization. Usually, a victim of phishing is not aware that the email sent contains malicious software or would redirect them to fraudulent websites tricking them into divulging information, be it personal or financial (account IDs or credit card details). In phishing, the attackers trick people into clicking a malignant link that would appear legitimate. Jang-Jaccard & Nepal (2014) explain how attackers are adopting increasingly sophisticated tools to phish and the need to address cybersecurity challenges. In addition, Alkhalil et.al (2021) discussed the five various phases carried out in the lifecycle of a phishing attack.

Pre-existing measures against phishing attacks include shutting down malicious websites by the Internet Service Provider (ISP) (Hutchings et al., 2016) and the use of warning tools embedded in browsers to indicate malicious sites once they are being accessed by the user. The evolution of phishing attacks has created techniques that prey on the vulnerability of both the computer systems and users. Therefore, researchers need to develop proactive measures to tackle this menace (Lim et al., 2020).

In view of the above, this paper addresses the following research questions: Which category of features gives better prediction of phishing URLs? Which classification

*Corresponding Author

Section B- ELECTRICAL/ COMPUTER ENGINEERING & RELATED SCIENCES
Can be cited as:
Igwilo C.M. and Odumuyiwa V. (2022): Comparative Analysis of Ensemble Learning and Non-Ensemble Machine Learning Algorithms for Phishing URL Detection, *FUOYE Journal of Engineering and Technology* (FUOYEJET), 7(3), 305-312. http://doi.org/10.46792/fuoyejet.v7i3.807

model algorithm gives higher accuracy score in detecting phishing URLs? This paper's contribution includes detecting phishing URLs using both ensemble learning and non-ensemble machine learning algorithms; and performing a comparison between the accuracy of ensemble learners against other machine learning classifiers used. On one hand, stacking classifier, Random Forest, Bagging classifier, Gradient boosting, and Adaboost algorithms were used for ensemble learning. Whereas Decision tree, K-Nearest Neighbours and Logistic Regression were used as the non-ensemble learners.

2 Literature Review

2.1 Related Work

One of the most predominant methods of cybercrime is phishing. It was first discussed in a newsletter in 1996 after an attack on American Online (AOL) accounts (Ollmann, 2008). According to (Verizon: 2019 Data Breach Investigations Report, 2019), phishing amounts to 78% of all Cyber-Espionage. Widup et.al. (2018) reported that 22% of phishing victims clicked on the phishing links sent and only 17% reported the incident. Phishing detection approaches are categorized into non-classification and classification approaches in relation to feature variation and machine learning. The use of White lists, Black lists and heuristics detection methods are considered non-classification approaches while classification approaches include machine learning techniques.

Phishing detection systems that use whitelists create list of legitimate websites that supply the necessary information, while blacklists contain phishing URLs. Every website that is not on the whitelist is flagged as potentially dangerous. Han et al. (2012) proposed a solution that allows the system to defend against phishing attacks by combining visual similarity-based techniques and white lists. Jain & Gupta (2016) used a strategy that uses an automatically updated white-list of legitimate websites to notify web users of malicious URLs. This work obtained 86.02% accuracy score. Le et al. (2018) discuss the works done during the early stages of malicious URL detection where blacklisting, regular expression, and signature matching were mostly used for URL detection. The problem these models faced was that they were unable to detect new URLs and it required that the database used be updated regularly. Due to these challenges, machine learning algorithms were introduced to detect malicious and phishing URLs efficiently.

Over the years, experiments carried out using machine learning (ML) show that ML techniques can effectively be used in developing anti-phishing tools (Abdelhamid et al., 2017). A lot of literatures (Sahingoz et al., 2019), (Ubing et al., 2019) discuss the use of classification algorithms like K-Nearest Neighbours (also known as KNN), Artificial Neural Networks (ANN) and Decision Tree (DT) as a strategy to mitigate phishing attacks. Using ML algorithms requires that feature extraction be performed on the dataset. Manually extracting features from URLs requires extensive domain knowledge of the URLs. However, feature engineering approaches in ML can be used to extract good features from URLs. Sahoo et al. (2019) used lexical features (Ma et al., 2009b) host-based features(Ma et al., 2009a), blacklist features (Felegyhazi et al., 2010), content features (Canali et al., 2011), and popularity features as a combination of features used in the classification model (Cao et al., 2016). The blacklist features were used to predict the presence of a URL in a blacklist database by Felegyhazi et al. (2010); string properties of a URL were used to get lexical features by Ma et al. (2009b). The length of URL, the number of redirections and the presence of '@'' symbol in URL were some of the lexical features extracted. Hostname properties of the URL, such as, IP Address, WHOIS information, geographic location, were used to get the host-based features. Content features are Information related to popularity scores, ranking, and source of sharing which are extracted from HTML and JavaScript when a user accesses a website through a malicious URL (Le et al., 2018).

Preethi & Velmayil (2016) introduced a PrePhish method which allows real-time URL phishing detection. The extracted features from the sample dataset used by the author are analysed using the Support Vector Machine (SVM), Random Forest (RF) and Naïve Bayes (NB). The authors achieved 97.83% accuracy for correctly classified phishing URLs. The limitation of this research was that the PrePhish algorithm relied on URL lexical analysis only in detecting malicious URLs. Baykara & Gürel, (2018) proposed a model called an Anti-phishing Simulator that examines the content of an email to detect spam emails. The model was built using the Bayesian classifier and each word stored has weights assigned to it, with spam and harmful words given higher weights. In this method, the user is protected without having to open the email using keywords stored in the database of the Bayes network. The limitation was that the simulator depended on the database of spam words provided which may be limited in vocabularies. (Mohammada, Shitharthb & Kumarc, 2020) used artificial neural network anti-phishing concept in his work, this model successfully determines whether the phishing email is known phishing or unknown phishing. To improve URL categorization, the Feed-Forward Backpropagation and Levenberg-Marquart methods of Artificial Neural Networks (ANN) are used, along with a fuzzy inference system to produce results using sparse social feature data. They claimed that their model can distinguish between known and unknow email phishing via URL with accuracy.

Ubing et al. (2019) used feature extraction method which contributed to improving the accuracy of phishing website detection. A 95% accuracy rate was observed, proving that feature selection algorithm could be more effective at detecting URL phishing. Sahingoz et al. (2019) had a 97.98% accuracy rate for detecting malicious URLs when NLP based features and a total of seven classification algorithms were used. The authors developed a methodology based on malicious URLs from PhishTank datasets. Naïve Bayes, Decision Tree (DT), K-Nearest Neighbour (KNN), Adaboost (AB), Random Forest, Sequential Minimal Optimization (SMO) which is a fast-training method for SVM, and K-Star were the

http://doi.org/10.46792/fuoyejet.v7i3.807 http://journal.engineering.fuoye.edu.ng/

seven classifiers employed. Due to the volume of the datasets used, their work recorded a slow execution rate. Bahnsen et al. (2017) utilized Long Short-Term Memory (LSTM) and recurrent neural network (RNN) for URL detection where both performed admirably. It was observed that LSTM outperformed the single machine learning models. Vinayakumar et al. (2019) made a comparison between machine learning with feature engineering methods and deep learning with character level embedding for URL phishing detection. Their model took a long time to train and the same applies to all other deep learning approaches. Pandey & Chadawar (2022) developed a hybrid ensemble model to determine whether or not a URL is safe to utilize. The hybrid model integrated MLP (3 weak learners), SVM (4 weak learners), decision trees (5 weak leaners), and the random forests (5 weak learners). This model achieved an accuracy of 85.37%.

This paper seeks to determine which category of features (lexical, DNS, HTML and tokenization feature) produces better accuracy score. The paper also compares the accuracy score between non-ensemble learners and ensemble learners.

2.2 Ensemble Learning Techniques

2.2.1 Bagging

Bagging is obtained from bootstrap aggregating used in ensemble system for machine learning classification algorithm. Bootstrapped samples of the training data are used to obtain diversity of classifiers in bagging. That is, various training data subsets are picked at random from the full training dataset – with replacement. Each subset of training data is utilized to train a particular classifier (Polikar, 2009). Each classifier votes to obtain an outcome of the model.

2.2.2 Boosting

Boosting is an ensemble strategy that improves predictions by learning from the mistakes of preceding predictors. In boosting, weak learners are arranged in sequence, therefore, allowing weak learners to learn from the next to create better predictive models. Boosting can be Adaptive Boosting (AdaBoost), and Gradient Boosting (GB). In AdaBoost, all records are assigned sample weights based on the classifier's performance. At the end of the first classification, the data that were incorrectly predicted are given higher weights and priority, and then delivered as input to the next model generated in a sequential order. Gradient Boosting (GB) is based on three major components: a loss function, a weak learner, and an additive model. The learners (usually decision tree) are linked to reduce the preceding tree's errors, the loss function identifies residuals, and the additive component originates from the fact that trees are added to the model over time, causing existing tree values to change. Gradient descent optimization is used to reduce the error between specified values. In order to limit the inaccuracy, the weights are only adjusted after the mistake has been calculated. The output of the new tree is then added to the output of the previous trees in the model. This method is repeated until a predetermined number of trees have been achieved or the loss has been minimized below a particular level. The learning technique in gradient boosting fits new models to offer a more accurate estimate of the response variable. Because of its high flexibility, the GB can be tailored to any data-driven task. (Zhou, 2012)

2.2.3 Stacking

This ensemble method is often referred to as stacked generalization. This strategy works by allowing a training algorithm to aggregate the predictions of a number of other learning algorithms that are similar. In addition to selecting multiple sub-models, stacking allows the combination of an extra model known as the meta-classifier to allow the combination of the feature vectors to be trained again (Zhou, 2012).

3 Methodology

This section discusses the ensemble learning approach for improving phishing attempt detection on URLs. Steps undertaken in this work include data collection, data pre-processing and cleaning, feature extraction, tokenization of URLs and data classification. The raw dataset is pre-processed and prepared for classification algorithms to evaluate their performance. The dataset is prepared by extracting important features that aid in differentiating phishing websites from benign ones.

3.1 Proposed Work Flow

In this section, we described our ensemble approach to improve the detection of phishing attempts on websites. The steps taken are as shown in Figure 1. The raw dataset is pre-processed and prepared for classification algorithm to evaluate its' performance.

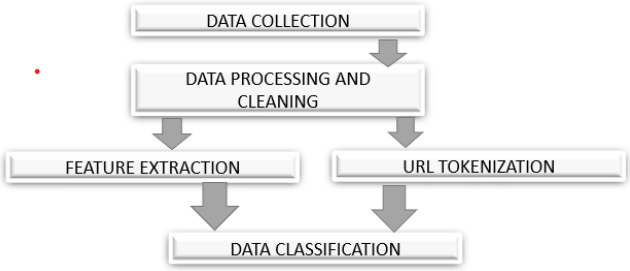

Fig. 1: Proposed Workflow

The proposed bagging model in this experiment, as shown in Figure 2, splits the training data into 3 subsets. Each training subsets is trained on DT. Each classifier gives a prediction which is then combined with majority voting to give an output. The classifier that gets the highest vote is chosen as the final outcome.

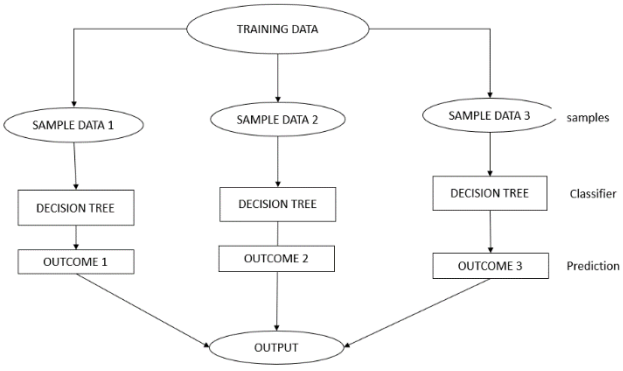

Fig. 2: Proposed Bagging Architecture

http://doi.org/10.46792/fuoyejet.v7i3.807 http://journal.engineering.fuoye.edu.ng/

The stacking model in this experiment uses Random Forest (RF) and KNN as level one classifiers and Logistic Regression is used as the meta-classifier. It is however important to note that RF is also an ensemble learner on its own but can be used in other ensemble learning architectures. Figure 3 shows the architecture of the stacking model.

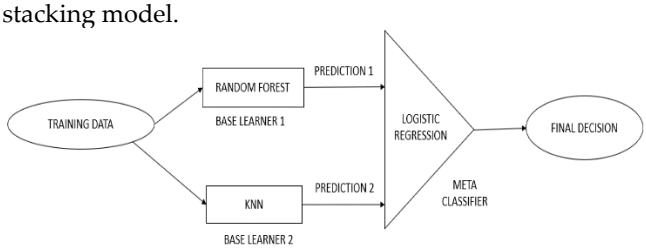

Fig. 3: Proposed Stacking Architecture

In boosting, each training subsets is trained sequentially. Each classifier gives a prediction which is then combined to give an output as shown in Figure 4.

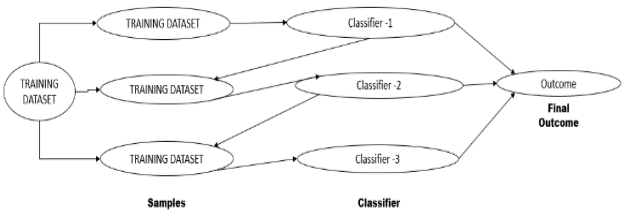

Fig. 4: Proposed Boosting Architecture

In this work, Adaboost and Gradient boosting were implemented using decision trees. Adaboost is implemented by connecting weak classifiers in such a way that the misclassified output of a weak classifier is fed into another weak classifier so that each weak classifier attempts to improve the classification of samples misclassified. The Decision Tree (weak classifier) that were employed were known as "stump." The correctly labelled samples had their weight reduced, while misclassified samples had their weight increased appropriately. The weight of a sample that was misclassified by the prior tree was increased, allowing the next tree to focus on accurately identifying the previously misclassified sample.

3.2 Pre-processing

The individual URLs are described by features (binary) which are grouped under Lexical features, Domain Named Based features, HTML Features, and tokenization of URL. The features were extracted from the URL strings in the datasets using functions. Data cleaning is done to remove all duplicate entries, fill in missing attributes or class values and remove the row of all missing class labels. An attribute mean was used to fill the values of a missing class since all extracted features were numeric (0 or 1).

3.3 Dataset

These experiments were conducted using two datasets. The first dataset is made up of 5000 Phishing URLs gotten from PhishTank (PhishTank, 2022) and 5000 legitimate URLs from University of New Brunswick (UNB, 2022.). The second dataset consists of 3000 legitimate and 3000 phishing URLs each gotten from Kaggle repository.

3.4 Feature Extraction

Feature extraction was performed on the dataset to extract important features of the URL. HTML and JavaScript-based features, Domain-based Features, and Lexical Features are gotten from the dataset. The target value 1 represents a phishing URL and 0 represents a legitimate URL. The generated dataset is passed through the classification algorithms used in this research.

3.5 URL Based Features

A breakdown of the features used and their description is provided in Table 1 while Table 2 provided some examples of URL and their classes based on the features.

Table 1. Description of used URL features

Features	Description	Value
Lexical Features		
Internet Protocol (IP) Address	The presence of an IP address or hexadecimal characters in the URL domain instead of using the domain name is related with 46.66% of phishing URLs. The presence of an IP address is indicated as 1 (phishing).	0 or 1
Presence of @ symbol in URL	If a URL contains '@', it causes the browser to disregard all previous characters before the symbol and focuses on the real address after the '@'. Any URL that contains '@' is assigned the value 1 (phishing).	0 or 1
Length of URL	A malicious URL is concealed within a lengthy URL. Any URL with length greater than 64 is assigned a target value of 1 (phishing).	0 or 1
Redirect Request	The position of "//" is determined by the presence of HTTP and HTTPS. If a URL has HTTP, then the position of "//" is in the sixth place while for a URL that has HTTPS the position of "//" is the seventh place. Any URL that does not conform to this, is assigned a target value 1 (phishing).	0 or 1
Prefix or Suffix "-" in Domain	If any URL in the dataset contains '-' this symbol in the domain part of the URL, the value assigned to this feature is 1 (phishing) or else 0 (legitimate).	0 or 1
Using URL Shortening Services "TinyURL"	URL shortening exposes the website to high-security risk by potential malware attacks. If a URL is using any shortening services not included in our database, that URL is assigned 1 (phishing).	0 or 1
Domain-Based Name (DNS) Features		
Domain Age	Phishing websites have a relatively brief lifespan. The WHOIS database was used to ascertain the domain's age. A domain age greater than1 year is regarded as benign URL.	0 or 1
Website Validity	The end period of the domain is used to assign a target value. A domain whose ending period is less than 8 months, is considered malicious.	0 or 1
Name Server Record	The WHOIS database was used to identify the record of a legitimate hostname. Any hostname not found is assigned 1 (phishing)	0 or 1
Web Traffic	In this project, a threshold value of 100000 is used to compare. If the domain ranks above 100,000, it is regarded a legitimate URL and assigned a value of 0.	0 or 1
HTML and JavaScript-based Features		
IFrame Redirection	Phishers can utilize the "iframe" tag to embed malicious URLs. Here, any URL with an Iframe is assigned 1 (phishing).	0 or 1

http://doi.org/10.46792/fuoyejet.v7i3.807 http://journal.engineering.fuoye.edu.ng/

Website Forwarding	The threshold value was assigned four therefore any URL that redirects to four or more webpages is termed phishing and a value of 1 is assigned to it.	0 or 1
Tokenization Features		
Tokenization of URL	These tokens are words or sub-words derived from the URL string by using website delimiters. The derived tokens are then used to prepare unique tokens in the vocabulary (corpus). Each vocabulary was tested as a feature using Count Vectorizer and TD-IDF approaches used in Natural Language Processing (NLP).	0 or 1

Table 2. Examples of used URL features

URL	Description	Value
http://93.186.251.133/exchange/signup/login.php	The presence of an IP address in the domain	1
http://br16.teste.website/~confi470/WWW.BRADESCO.COM.BR/shtm/desktop/home.php?cli=&/zNrAy09pKU/ibvjGfIEsD.php	The length of the URL is above 64	1
https://t.co/SingcAr1bM	URL shortening services not included in the database	1
http://santeassessoria.com/	No record of the domain in the Whois database	1

3.6 Experimentation

The experiment involves splitting the data into training and test sets using 70:30. The experiment was carried out using Scikit-learn and pandas. Each classifier is trained using a training set, and the performance of the classifier is evaluated using a testing set. The classifiers were executed twice and the average of both results are reported in Section 4. For the tokenization approach, the URLs were broken into tokens and used for classification process.

4 Results and Discussion

This section reports the results derived from the ML and Ensemble learners trained on the two datasets collected and were implemented by testing the outcome of sample URLs. As described in the methodology, four different categories of features were used for feature extraction. The results gotten from using tokenization, lexical features, DNS features, and HTML features were compared to identify the best approach in phishing URL detection using performance measures such as Accuracy and Precision.

4.1 Evaluation Metrics

Accuracy, Precision, F1 score and Recall were employed to measure classification performance in these experiments. Accuracy is simply a ratio of correctly predicted observation to the total observations.

$$\text{Accuracy} = (TP+TN)/(TP+FP+FN+TN) \quad (1)$$

Precision represents the ratio of correctly predicted positive observations to total predicted positive observations.

$$\text{Precision} = TP/(TP+FP) \quad (2)$$

F1-score is a weighted average of precision and recall. It is an important performance metric to evaluate the overall performance of our method.

$$\text{F1 Score} = 2*(\text{Recall} * \text{Precision})/(\text{Recall} + \text{Precision}) \quad (3)$$

Recall is the ratio of correctly predicted positive observations to all the observations in actual class.

$$\text{Recall} = TP/(TP+FN) \quad (4)$$

Where True Positive (TP): URLs which are legitimate and are actually predicted legitimate.
True Negative (TN): URLs which are malicious and are actually predicted malicious.
False Positive (PN): URLs which are not-legitimate and are actually predicted legitimate.
False Negative (FN): URLs which are legitimate and are actually predicted not-legitimate.

4.2 Results

The experiments carried out showed improved accuracy score for ensemble learners when compared to other classification methods in this experiment. Balanced datasets with equal instances of malicious and legitimate URLs were used for training and testing set respectively. The training set with the extracted features was given as input to all the algorithms. Tables 3 to 6 summarise the outcomes of our experimentations.

4.3 Discussion

Table 3 shows the accuracy score of test data on non-ensemble classifiers and the ensemble learners when using lexical features alone on datasets 1 and 2 as well as when using all the three features on both data sets. Using lexical features on Dataset 1, the DT classifier gives the highest accuracy score amongst the three single classifiers (non-ensemble learners) with an accuracy rate of 82%, while logistic regression and KNN record 81.9% and 79% accuracy respectively and the Stacking classifier has the highest accuracy score amongst other ensemble learners with an accuracy score of 84% for dataset 1. The gradient boosting classifier has the least accuracy score amongst the ensemble learners used with an accuracy score of 80.5%. In comparison to other ensemble learners, this classifier does not perform well for phishing detection.

Using lexical features alone on dataset 2, the non-ensemble classifier with the highest accuracy score was the decision tree with a score of 84.9% while the Logistic Regression was the lowest with 69.1%. The stacking classifier gave the best results amongst all classifiers with an accuracy score of 85.3%. When lexical, DNS, and HTML features retrieved from the URL are integrated, the results obtained from both datasets show that a combination of all three features give higher accuracy score as compared to using just one feature. On dataset 1, the KNN and Stacking classifier both produced an accuracy score of 90% which was higher than other models. While on dataset 2, the accuracy score of the stacking classifier, which performed better than other models, was 92.5%.

http://doi.org/10.46792/fuoyejet.v7i3.807 http://journal.engineering.fuoye.edu.ng/

Table 3. Result of classifiers using lexical features on dataset 1 and 2

Models	Using lexical features for dataset 1				Using lexical features for dataset 2				Using all the three features for dataset 1				Using all the three features for dataset 2			
	Accuracy (%)	Precision (%)	F1 score (%)	Recall (%)	Accuracy (%)	Precision (%)	F1 score (%)	Recall (%)	Accuracy (%)	Precision (%)	F1 score (%)	Recall (%)	Accuracy (%)	Precision (%)	F1 score (%)	Recall (%)
Decision Tree (DT)	82	74	84	96	84.9	85	84	83	88.3	85	87	89	91.6	93	92	91
Logistic Regression	81.9	73	83	96	69.1	63	72	85	88.3	82	89	98	91.6	93	92	91
KNN	79	75	80	85	83.8	78	84	91	90	86	91	97	92	93	92	91
AdaBoost	81.4	74	84	96	85.1	86	84	83	88.3	85	89	94	91.5	93	92	91
Bagging	82.2	74	84	96	84.8	82	85	88	86.7	79	87	98	91.5	93	91	90
Gradient Boosting	80.5	74	83	95	85.1	86	84	83	88.3	85	89	94	90	92	91	90
Random Forest	82.4	75	84	96	85.1	89	84	83	89	84	91	100	91.6	93	92	91
Stacking	**84**	75	84	96	**85.3**	85	84	84	**90**	86	91	97	**92.5**	93	92	91

As may be seen in tables 4 to 5, we recorded an improved performance using both non-ensemble and ensemble learning classifiers on tokenized URL. Across all of the classifiers employed in this research, the tokenization feature produced best results. For dataset 1, the stacking classifier has highest accuracy score with 96% and lowest accuracy score of 86.7% was gotten from the DT classifier. Table 5 illustrates the outcomes of the stacking classifier recording 99.3% and decision tree recording 93% as the highest and lowest accuracy score respectively on dataset 2. We also experimented with domain-based features separately and observed that using such features alone performed badly and Table 6 shows that the accuracy score was poor. It implies that this feature is not very efficient in phishing detection. Having established that the ensemble learners performed better than the non-ensemble learners, Figures 5 and 6 provide a plot for visual comparison of ensemble learners performance utilizing combination of lexical, HTML and DNS features on dataset 1 to their performance on the same dataset using URL tokenization. The stacking classifier on tokenization feature gave the best outcome.

Table 4. Result of classifiers using tokenization on dataset 1

Model	Accuracy (%)	Precision (%)	F1 score (%)	Recall (%)
Decision Tree	86.7	93	87	82
Logistic Regression	93.5	93	94	96
KNN	95.5	95	95	96
AdaBoost	95	95	95	95
Bagging	95.5	95	95	96
Gradient Boosting	93.8	95	95	95
Random Forest	95.8	96	96	96
Stacking	**96**	97	96	96

Table 5. Result for classifiers using tokenization on dataset 2

Model	Accuracy (%)	Precision (%)	F1 score (%)	Recall (%)
Decision Tree	93	98	93	88
Logistic Regression	98.9	99	99	99
KNN	97.4	98	97	97
AdaBoost	98.6	99	99	99
Bagging	99	100	99	99
Gradient Boosting	98.6	99	98	98
Random Forest	97	99	97	95
Stacking	**99.3**	99	99	99

Table 6. Average result of DNS feature from datasets 1 and 2

Model	Accuracy (%)
AdaBoost	50
Bagging	50
Gradient Boosting	50
Random Forest	51
Stacking	53
Decision Tree	50
Logistic Regression	52
KNN	49

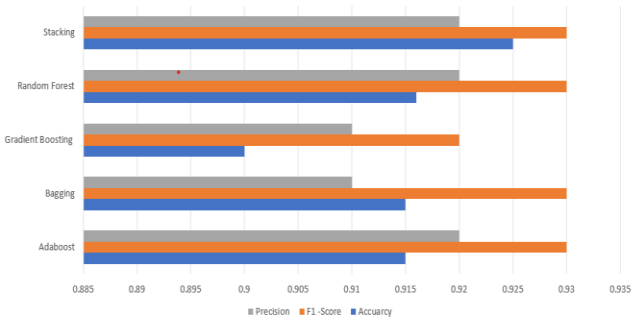

Fig. 5: Result of ensemble learners using all three features (DNS, HTML, LEXICAL features) on Dataset 2.

http://doi.org/10.46792/fuoyejet.v7i3.807 http://journal.engineering.fuoye.edu.ng/

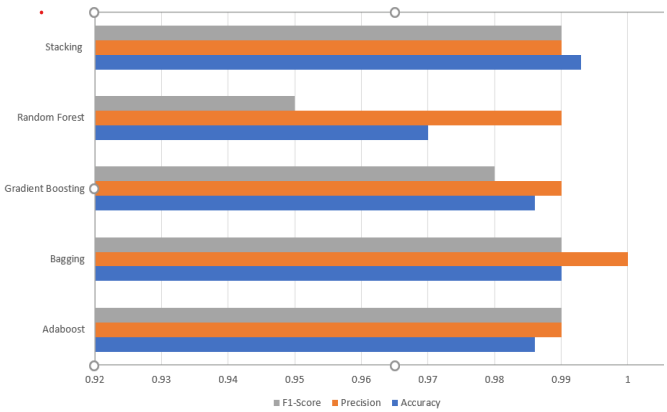

Fig. 6: Result of URL tokenization using ensemble learners on Dataset 2

5 CONCLUSION

This research attempted determining a better way of detecting URL-based phishing attacks by experimenting with 3 non-ensemble classifiers and 5 ensemble learner classifiers on two different balanced phishing datasets. Results obtained show the superior capability of stacking ensemble learner in URL phishing detection when URL tokenization method is used as a feature extraction strategy. The stacking model proposed in this experiment uses Random Forest and KNN as level one classifiers and Logistic Regression as a meta-classifier.

REFERENCE

Abdelhamid, N., Thabtah, F., & Abdel-jaber, H. (2017). Phishing detection: A recent intelligent machine learning comparison based on models content and features. *In 2017 IEEE international conference on intelligence and security informatics (ISI)*, 72-77. IEEE. https://doi.org/10.1109/ISI.2017.8004877

Adepetun. A. (2019, June 27). Nigeria's exposure to phishing attacks rises as cybercrime cost hits $6 trillion | The Guardian Nigeria News - Nigeria and World News — Business — The Guardian Nigeria News – Nigeria and World News. https://guardian.ng/business-services/nigerias-exposure-to-phishing-attacks-rises-as-cybercrime-cost-hits-6-trillion/ Accessed on January 15, 2022

Alkhalil, Z., Hewage, C., Nawaf, L., & Khan, I. (2021). Phishing Attacks: A Recent Comprehensive Study and a New Anatomy. *Frontiers in Computer Science*, 3, 563060. https://doi.org/10.3389/FCOMP.2021.563060

Bahnsen, A. C., Bohorquez, E. C., Villegas, S., Vargas, J., & Gonzalez, F. A. (2017). Classifying Phishing URLs using Recurrent Neural Networks. *In 2017 APWG symposium on electronic crime research (eCrime)*, 1-8. IEEE. https://doi.org/10.1109/ECRIME.2017.7945408

Baykara, M., & Gürel, Z. Z. (2018). Detection of Phishing Attacks. *6th International Symposium on Digital Forensic and Security. ISDFS 2018 Proceeding*, 1-5. https://doi.org/10.1109/ISDFS.2018.8355389

Canali, D., Cova, M., Vigna, G., & Kruegel, C. (2011). Prophiler: A Fast Filter for the Large-Scale Detection of Malicious Web Pages Categories and Subject Descriptors. *Proc. of the International World Wide Web Conference (WWW)*, 197-206.

Cao, J., Li, Q., Ji, Y., He, Y., & Guo, D. (2016). Detection of Forwarding-Based Malicious URLs in Online Social Networks. *International Journal of Parallel Programming*, *44(1)*, 163-180. https://doi.org/10.1007/s10766-014-0330-9

Felegyhazi, M., Kreibich, C., & Paxson, V. (2010). On the potential of proactive domain blacklisting. *LEET*, *10*, 6-6.

Han, W., Cao, Y., Bertino, E., & Yong, J. (2012). Using automated individual white-list to protect web digital identities. Expert Systems with Applications. *Expert Systems with applications*, *39(15)*, 11861-11869. https://doi.org/10.1016/j.eswa.2012.02.020

Hutchings, A., Clayton, R., & Anderson, R. (2016). Taking down websites to prevent crime. *In 2016 APWG symposium on electronic crime research (eCrime)*, 1-10, IEEE. https://doi.org/10.1109/ECRIME.2016.7487947

Jain, A. K., & Gupta, B. B. (2016). A novel approach to protect against phishing attacks at client side using auto-updated white-list. *Eurasip Journal on Information Security*, *2016(1)*, 1-11. https://doi.org/10.1186/s13635-016-0034-3

Jang-Jaccard, J., & Nepal, S. (2014). A survey of emerging threats in cybersecurity. *Journal of Computer and System Sciences*, *80(5)*, 973-993. https://doi.org/10.1016/j.jcss.2014.02.005

Lumpur, K. (2020, January 9). 91% of all cyber attacks begin with a phishing email to an unexpected victim | Deloitte Malaysia | Risk Advisory | Press releases. https://www2.deloitte.com/my/en/pages/risk/articles/91-percent-of-all-cyber-attacks-begin-with-a-phishing-email-to-an-unexpected-victim.html Accessed on January 15, 2022

Le, H., Pham, Q., Sahoo, D., & Hoi, S. C. H. (2018). URLNet: Learning a URL representation with deep learning for malicious URL detection. *arXiv preprint arXiv:1802.03162*. Accessed on January 15, 2022

Lim, W. H., Liew, W. F., Lum, C. Y., & Lee, S. F. (2020). Phishing Security: Attack, Detection, and Prevention Mechanisms. *In Proceedings of the International Conference on Digital Transformation and Applications (ICDXA).*

Ma, J., Saul, L. K., Savage, S., & Voelker, G. M. (2009a). Beyond blacklists: Learning to detect malicious web sites from suspicious URLs. *In Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 1245-1254. https://doi.org/10.1145/1557019.1557153

Ma, J., Saul, L. K., Savage, S., & Voelker, G. M. (2009b). Identifying suspicious URLs: An application of large-scale online learning. *In Proceedings of the 26th annual international conference on machine learning*, 681-688. https://doi.org/10.1145/1553374.1553462

Odumuyiwa, V., & Chibueze, A. (2020). Automatic Detection of HTTP Injection Attacks using Convolutional Neural Network and Deep Neural Network. *Journal of Cyber Security and Mobility*, *9(4)*, 489-514.

Ogbonnaya, M. (2020, October 19). Cybercrime in Nigeria demands public-private action - ISS Africa. https://issafrica.org/iss-today/cybercrime-in-nigeria-demands-public-private-action Accessed on January 15, 2022

Ollmann, G. (2008) The Phishing Guide: Understanding and Prevent Phishing Attacks. *Security*, 1–42.

Pandey, A., & Chadawar, J. (2022). Phishing URL Detection using Hybrid Ensemble Model. *International Journal of Engineering Research & Technology*, *11(4)*, 479–482.

PhishTank Developer Information (2022). Retrieved February 17, 2022, from https://www.phishtank.com/developer_info.php

Polikar, R. (2009). Ensemble learning. *Scholarpedia*, *4(1)*, 2776. https://doi.org/10.4249/scholarpedia.2776 Accessed on January 15, 2022

Preethi, V., & Velmayil, G. (2016). Automated Phishing Website Detection Using URL Features and Machine Learning Technique. *International Journal of Engineering and Techniques*, *2(5)*, 107–115. http://www.ijetjournal.org

Sahingoz, O., Buber, E., Demir, Ö., & Diri, B. (2019). Machine learning based phishing detection from URLs. *Expert Systems with Applications*, *117*, 345–357. https://doi.org/10.1016/J.ESWA.2018.09.029

Sahoo, D., Liu, C., & Hoi, S. C. H. (2019). Malicious URL Detection using Machine Learning: A Survey. *arXiv preprint*

arXiv:1701.07179. Accessed on January 15, 2022

Mohammada, G.B., Shitharthb, S. & Kumarc, P.R., 2020. Integrated machine learning model for an URL phishing detection. International Journal of Grid and Distributed Computing, 14(1), pp.513-529.

Sikdar, B. (2020). Security and privacy for the internet of things. International Conference on Electrical Engineering, *In 2020 7th International Conference on Computer Science and Informatics (EECSI),* October 2020. https://doi.org/10.23919/EECSI50503.2020.9251914

Ubing, A. A., Kamilia, S., Jasmi, B., Abdullah, A., Jhanjhi, N. Z., & Supramaniam, M. (2019). Phishing Website Detection: An Improved Accuracy through Feature Selection and Ensemble Learning. *International Journal of Advanced Computer Science and Applications, 10(1),* 252-257

UNB (2022) URL 2016 Canadian Institute for Cybersecurity | Retrieved February 17, 2022, from https://www.unb.ca/cic/datasets/url-2016.html

Verizon: 2019 Data Breach Investigations Report. (2019). Computer Fraud & Security, 2019(6), 4. https://doi.org/10.1016/s1361-3723(19)30060-0

Vinayakumar, R., Soman, K. P., Prabaharan Poornachandran, Akarsh, S., & Elhoseny, M. (2019). Deep learning framework for cyber threat situational awareness based on email and URL data analysis. Advanced Sciences and Technologies for Security Applications, 87–124. https://doi.org/10.1007/978-3-030-16837-7_6

Vinayakumar R., Soman K.P., Prabaharan Poornachandran, Akarsh S., Elhoseny M. (2019) Deep Learning Framework for Cyber Threat Situational Awareness Based on Email and URL Data Analysis. *In Cybersecurity and Secure Information Systems,* 87-124. Springer, Cham. https://doi.org/10.1007/978-3-030-16837-7_6

Widup, S., Spitler, M., Hylender, D., and Bassett, G. (2018). 2018 Verizon data breach investigations report. Technical report. Available at https://www.researchgate.net/publication/324455350_2018_Verizon_Data_Breach_Investigations_Report Accessed on January 15, 2022.

Zhou, Z.H. (2012). *Ensemble Methods: Foundations and Algorithms. (pp. 23 - 57).* CRC Press. https://tjzhifei.github.io/links/EMFA.pdf