Human-Intelligent Systems Integration
https://doi.org/10.1007/s42454-024-00053-9

ORIGINAL PAPER

Intelligent analysis to detect phishing websites using machine learning ensemble techniques

Mithilesh Kumar Pandey[1] · Rekha Pal[1] · Saurabh Pal[1] · Alok Kumar[2] · Arvind Kumar Shukla[3] · Dhyan Chandra Yadav[4]

Received: 13 January 2023 / Accepted: 17 August 2024

Abstract
The Internet has grown to be a vital part of our everyday existence. Web browsing is the most popular Internet service. A lot of people use their browser for banking, online shopping, bill paying, and mobile phone recharging. Due to the extensive use of this service, users are exposed to many security risks, including cybercrime. One kind of online danger that lures consumers into connecting with a phoney website is cyber phishing. This study paper's primary objective is to safeguard sensitive user data. The suggested model is created in three stages. In the first phase, we select a dataset to train on and subsequently use the dataset to test classifiers. After applying the three classifiers in step 2 and finishing all of the predictions in step 3, we found that XGBoost performed better than the machine learning techniques AdaBoost and Gradient boosting.

Keywords Machine learning classifiers: XGB · AdaBoost · Gradient boosting · Pearson correlation · Phishing complex dataset

1 Introduction

Phishing is a fraudulent operation that uses social engineering and technology tools to get credentials, usernames, passwords, and account numbers from customers in order to obtain their personal information and bank account data. Social engineering is a common tactic used in phishing attempts to trick the target into clicking on a fake link that leads to a fraudulent webpage. The phoney website is made with the same process as the authentic one (Curtis et al. 2018). As a result, the attacker's server will receive the victim's request rather than the legitimate web server. Despite the fact that modern firewalls, anti-virus programmes, and specialist software cannot completely prevent online spoofing attacks, phishing still costs Internet users billions of dollars every year. The adoption of Secure Socket Layer (SSL) or a digital certificate does not protect online users from this type of attack (Chiew et al. 2018).

The following are the different forms of phishing attacks:

- Phishing assaults on websites usually begin with the creation of a phoney website since users of the Internet rely on a website's design to recognise it.
- A vast number of recipients get mass emails from the attacker warning them about a variety of scams, including the need to verify account information, fictitious charges, unauthorised account updates, system errors requiring users to re-enter their information, and new free services that require immediate action.
- These attacks appear covertly when consumers try to check in to a reliable website. After obtaining the user's credentials, the attacker sends them locally to the phishers.
- In order to deceive the user into providing personal information to the attacker, the attacker in this technique substitutes fake content for portions of the content seen on the legitimate website (Curtis et al. 2018).

1.1 Machine leaning environment

- Whenever someone opens an email and clicks on one of its links, or whenever they browse the Internet, the

✉ Saurabh Pal
drsaurabhpal@yahoo.co.in

[1] Department of Computer Applications, VBS Purvanchal University, Jaunpur, Uttar Pradesh 222001, India

[2] Department of Computer Applications, Vivek College of Education, Bijnor, India

[3] School of Computer Science & Applications, IFTM University, Moradabad, India

[4] Department of Computer Science and Engineering, MUIT, Lucknow, India

Published online: 14 September 2024

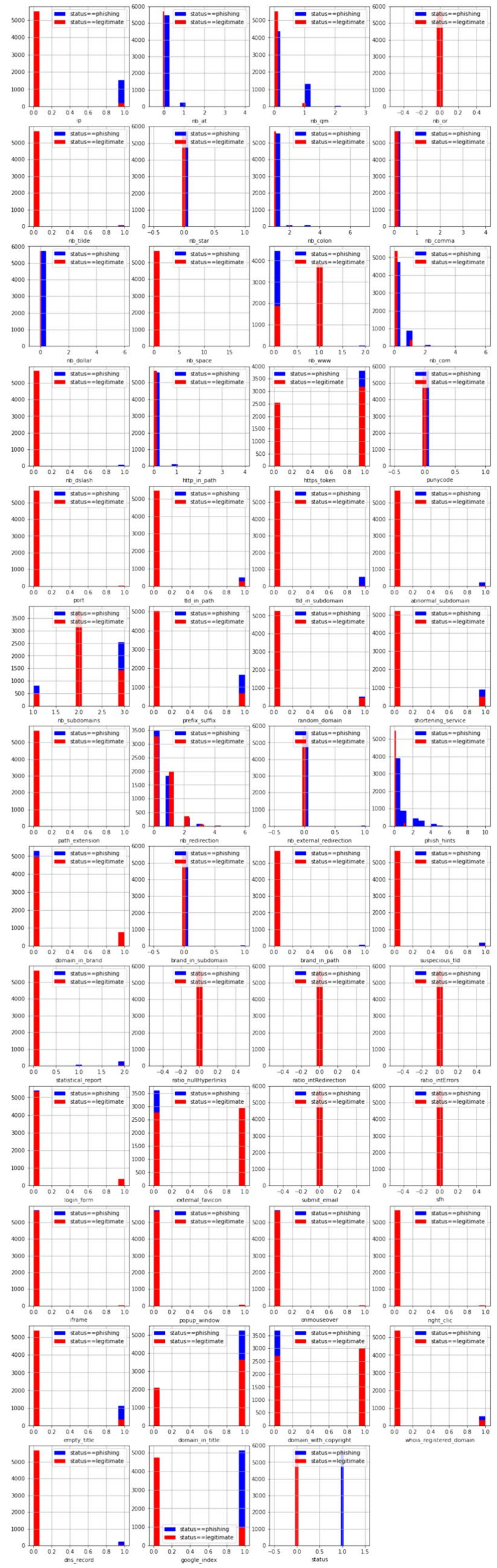


Fig. 1 Phishing dataset visualisation

user will be sent to a website that is either legitimate or not. This webpage, therefore, is simply test data.

- Once within the browser, a PHP script starts processing the test data to extract the features and save them in a data structure. Next, using rules it has learnt from other websites, the intelligent model will work within the browser to predict the kind of website.
- Finally, using similarity between attributes, the classifier's rules will be utilised to predict the type of test data.
- If the website they visited is found to be authentic, nothing will happen. On the other hand, in the event that the website proves to be fraudulent, the user will be warned of his danger using clever techniques (Alkawaz et al. 2020).

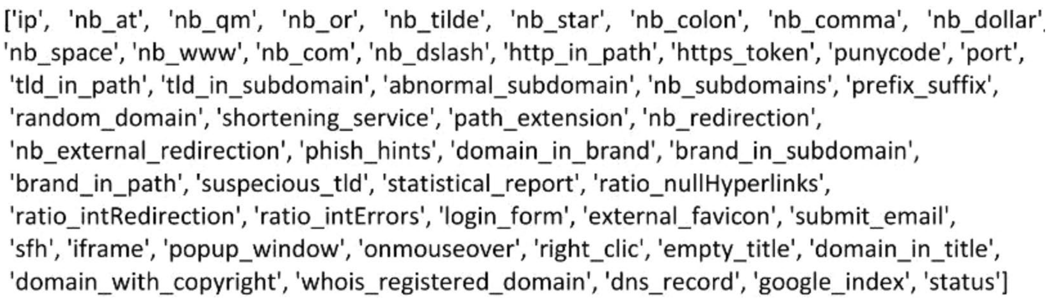
2 Related work

Méndez et al. used preprocessing techniques such tokenization, stemming, and stop-word removal for email text corpus datasets. They subsequently improved classification accuracy by using support vector machine (SVM) (Méndez et al. 2005). Ruskanda is summarised using stemming, lemmatization, stop-word removal, and noise removal. They selected the Ling-spam corpus dataset, which included 962 spam and ham communications in total. The authors used Naïve Bayes (NB) and support vector machine (SVM) to enhance prediction. In the end, scientists found that NB performed better than SVM (Ruskanda 2019). Alauthman experimented with data normalisation and discretization methods using the Twitter dataset. They used neural networks (NN), SVM, and random forests (RF) to increase classification accuracy. Over the course of the trial, the authors found that random forest yielded the best classification accuracy, at 84.30% (Alauthman 2020). Jain et al. choose tokenization and segmentation as their two data preparation techniques using datasets of 1.5 million posts from real-time Facebook data. The authors used RF classifiers in addition to SVM, NB, and RF for better outcomes. They found that random forest's F-measures values were calculated higher when compared with other classifiers (Jain et al. 2018). Ahmad et al. performed stop-word removal and stemming on the dataset. The authors used multilayer perceptron (MLP), NB and RF, and SVM algorithms on the two million tweets in the Honeypot dataset—both spam and non-spam—to achieve higher levels of accuracy and precision. They came at the conclusion that random forest generated precision and accuracy of 0.98 and 0.96, respectively (Ahmad et al. 2021). Inuwa et al. organised the Honeypot dataset, whereas SPD annotated the spam dataset both automatically and manually. The authors used support vector machine (SVM), random forest

(RF), multi-layer perception (MLP), maximum entropy, and gradient boosting to enhance prediction. They found that the proposed feature set increases system accuracy and that real-time spam detection is achievable, although there are limitations, such having to deal with the availability of lengthy tweets related to spamming operations. Finally, they reached 97.71% accuracy. Finally, they reached 97.71% accuracy, 99% precision, 97% recall and 98% is the F-score. (Inuwa-Dutse et al. 2018). Aiyar and Shetty made use of the 13,000-comment dataset from the YouTube channel. The authors used a range of machine learning techniques, such as SVM, RF, and Naive Bayes (NB) using N-grams-based features. The selected techniques enhanced Word2Vec classification accuracy and yielded the highest F1-score of 0.97 (Aiyar and Shetty 2018). A dataset including 131,000 tweets from 774 spam operations was compiled by Chu and associates. The authors used a variety of classification methods, such as KStar, Bayes Net, random forest, decision trees (DT), decision tables, random trees, and simple logistic content. They calculated the following categorization for the spam dataset: FPR-4.1% FNR-6.6% and 94.5% is the accuracy (Chu et al. 2012). Alharthi et al. collected a dataset of 10,000 Arabic tweets for their prediction. The authors used lengthy short-term memory machine learning techniques along with word embedding feature representation. The accuracy of the system's classification is influenced by the length of the tweet; the estimated values for accuracy, precision, and recall are 0.97, 0.98, and 0.95, respectively (Alharthi et al. 2021). Liu, Pang, and Wang used a dataset of 31,317 hotel ratings and 97,839 restaurant reviews for their classification. They used machine learning techniques, Bi-LSTM, and multimodal neural network models to investigate the beneficial features and improve performance, and they were able to get recall 0.80, precision 0.82, and F1-score 0.81 (Liu et al. 2019). Saidani et al. organised a dataset from the Enron Corpus that includes 2893 messages with 2412 hams and 481 ham phrases for better machine learning prediction. They came to the conclusion that it was important to update and improve the semantic qualities after conducting all of the studies: accuracy 0.98, recall 0.98, accuracy 0.98, and F1 measure 0.97 (Saidani et al. 2020).

3 Methodology

Data gathering is the first step in the implementation process. The dataset phase is crucial in order to guarantee error-free findings. The details will aid in the explanation and clarification of phishing as well as lawful activity. All of the characteristics that were retrieved from the UCI repository are shown in Fig. 1. A compilation of 57 phishing websites' attributes has been made. The categorical variables in the recovered dataset, "Legitimate" and "Phishy", have had their numerical values replaced with "1" and "– 1", respectively.



```
['ip', 'nb_at', 'nb_qm', 'nb_or', 'nb_tilde', 'nb_star', 'nb_colon', 'nb_comma', 'nb_dollar',
'nb_space', 'nb_www', 'nb_com', 'nb_dslash', 'http_in_path', 'https_token', 'punycode', 'port',
'tld_in_path', 'tld_in_subdomain', 'abnormal_subdomain', 'nb_subdomains', 'prefix_suffix',
'random_domain', 'shortening_service', 'path_extension', 'nb_redirection',
'nb_external_redirection', 'phish_hints', 'domain_in_brand', 'brand_in_subdomain',
'brand_in_path', 'suspecious_tld', 'statistical_report', 'ratio_nullHyperlinks',
'ratio_intRedirection', 'ratio_intErrors', 'login_form', 'external_favicon', 'submit_email',
'sfh', 'iframe', 'popup_window', 'onmouseover', 'right_clic', 'empty_title', 'domain_in_title',
'domain_with_copyright', 'whois_registered_domain', 'dns_record', 'google_index', 'status']
```

3.1 Machine learning techniques

Machine learning algorithms: The comma-separated values (CSV) file format was utilised in this investigation. The input file was examined by the Java and Python programmes. Three different machine learning classification techniques were employed to categorise the URL from the input URL collection.

Adaboost: Let's once again see all the steps taken in AdaBoost.

1. Generate system and make predictions.
2. Marked large weights to miss-classified points.
3. Generate next system.
4. Repeat steps 3 and 4.
5. Make a final model using the weighted average of individual models (Solomatine et al. 2004).

Gradient boosting algorithm:

1. Evaluate average of the target values from target label
2. Calculate the residuals values from proceeding formula:

   $[Residual = Actual\ value - Predicted\ value]$

3. Now construct a decision tree with the goal of predicting the residuals.
4. The target level values predicted by trees within the ensemble.
5. Calculated new residuals values.
6. Repeat steps 3 to 5 until the number of iterations matches the number specified by the hyperparameter (i.e., number of estimators).
7. Once trained, use all of the trees in the ensemble to make a final prediction as to the value of the target variable (Bikmukhametov and Jäschke 2019).

XG boost algorithm:

1. Generate a single leaf tree.
2. Calculate residuals values from prediction in previous tree.
3. Calculate the similarity score from appropriate node.
4. Also calculate Information gain.
5. Generate tree of desired length by previous pattern.
6. Predict the residual values by decision tree.
7. Calculate new residuals values.
8. Go back to step 1 and repeat the process for all the trees (Chen et al. 2021).

3.2 Proposed method

The dataset for this paper was obtained using three separate techniques. Figure 2 displays the model phishing dataset that has been recommended. Machine learning classifiers are used in it. Classifiers are trained using the first and second datasets and subsequently tested using the third combined dataset. The three components of the phishing detection system finally compare the results. In Figs. 2, 3, 4, 5, 6, 7, and 8, we examined the model's performance via three distinct lenses. The best prediction performance was achieved by XGBoost, which was followed by gradient boosting and Adaboost. Using a feature selection technique, the most important traits for successful phishing website detection are selected in this study. This is done to make sure that every legitimate website and phishing website has a distinct layout. The machine learning classifier that has been trained and evaluated comes next. The output of the machine learning classifiers will be the basis for the phishing detection model. Finally, this approach was used to test phishing websites.

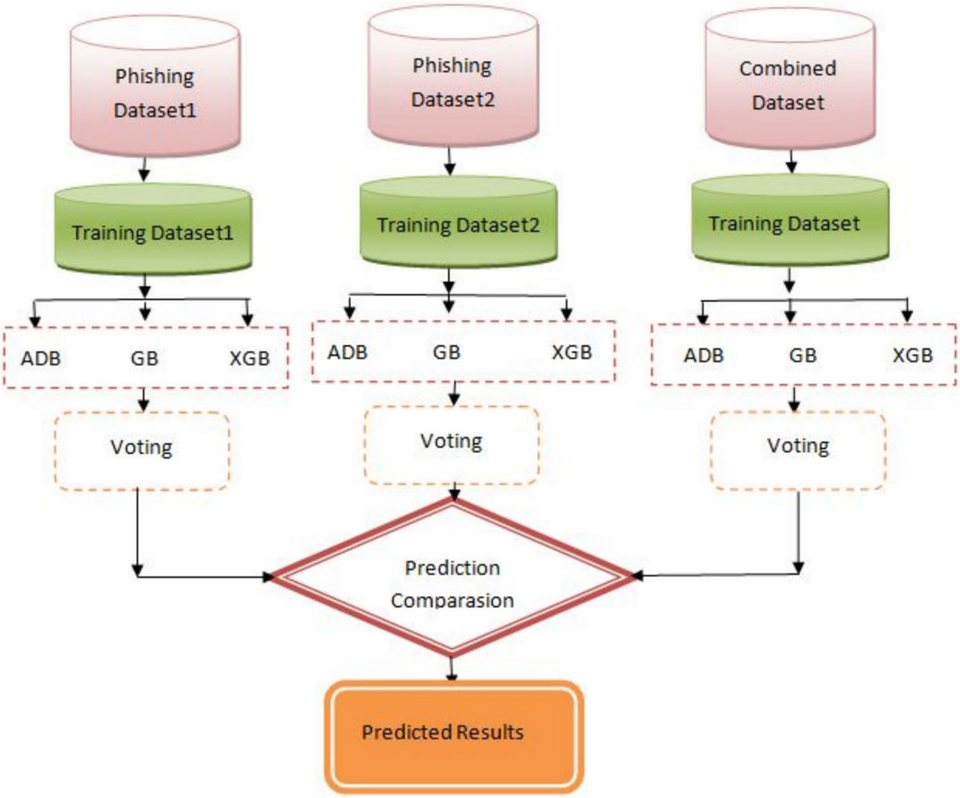


Fig. 2 Proposed model phishing dataset using machine learning classifiers

Fig. 3 First phishing dataset by Pearson correlation

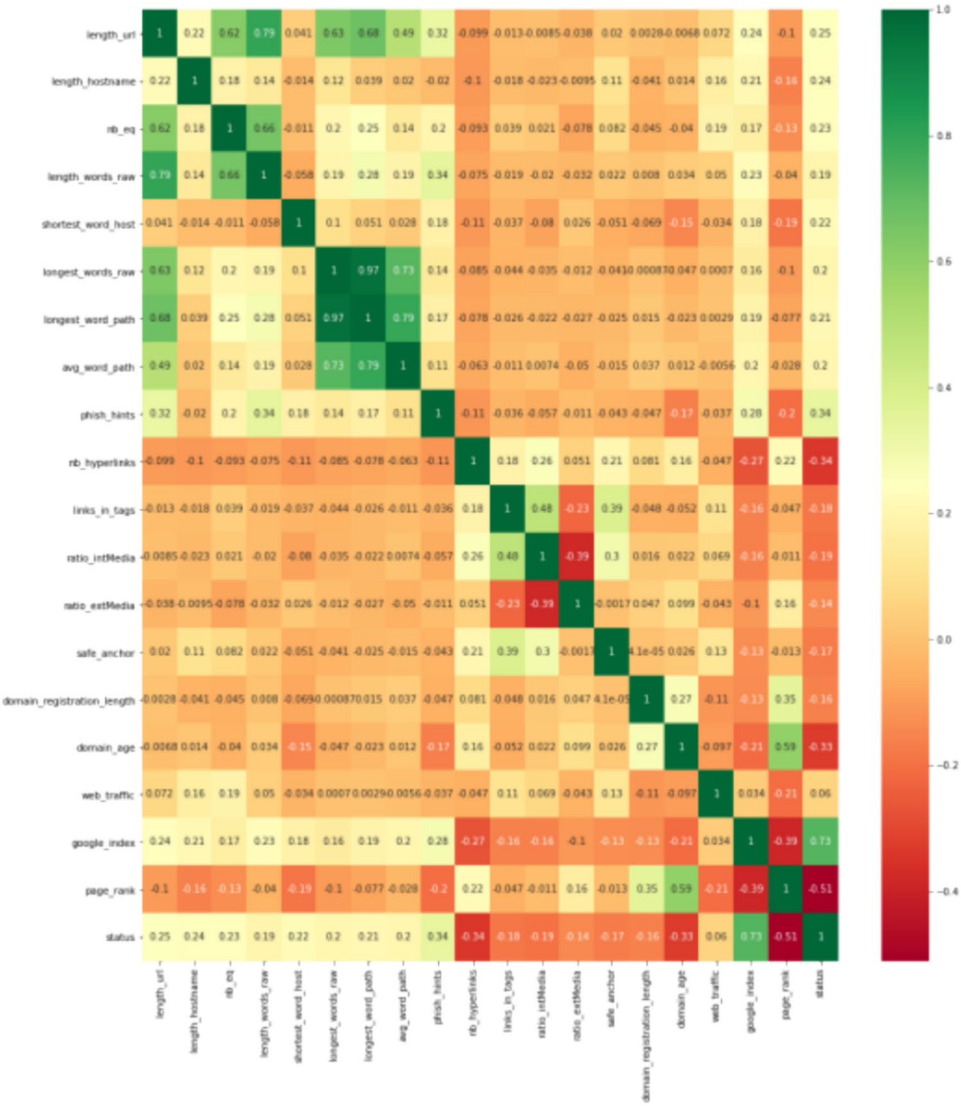
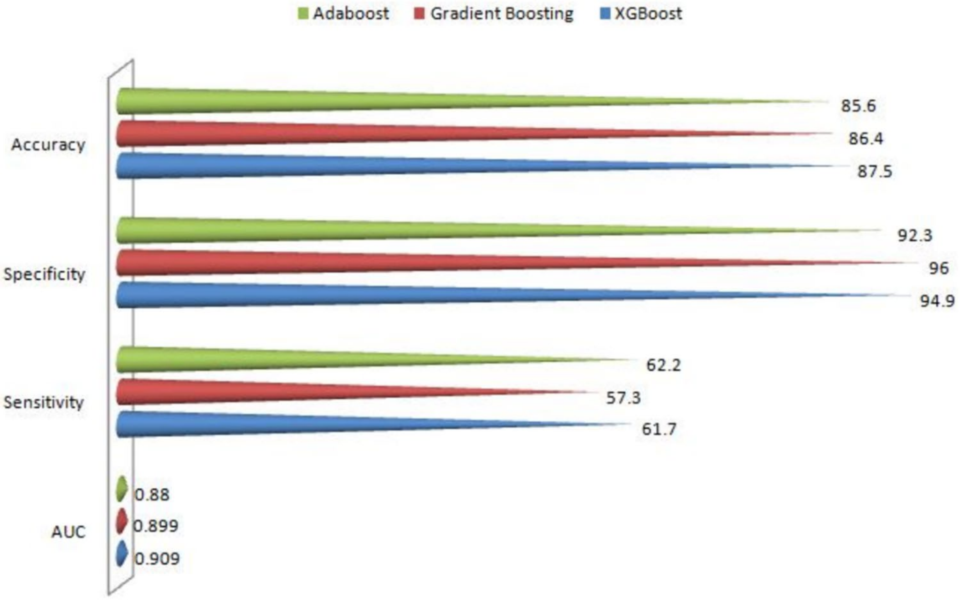


Fig. 4 First phishing dataset by boosting algorithms



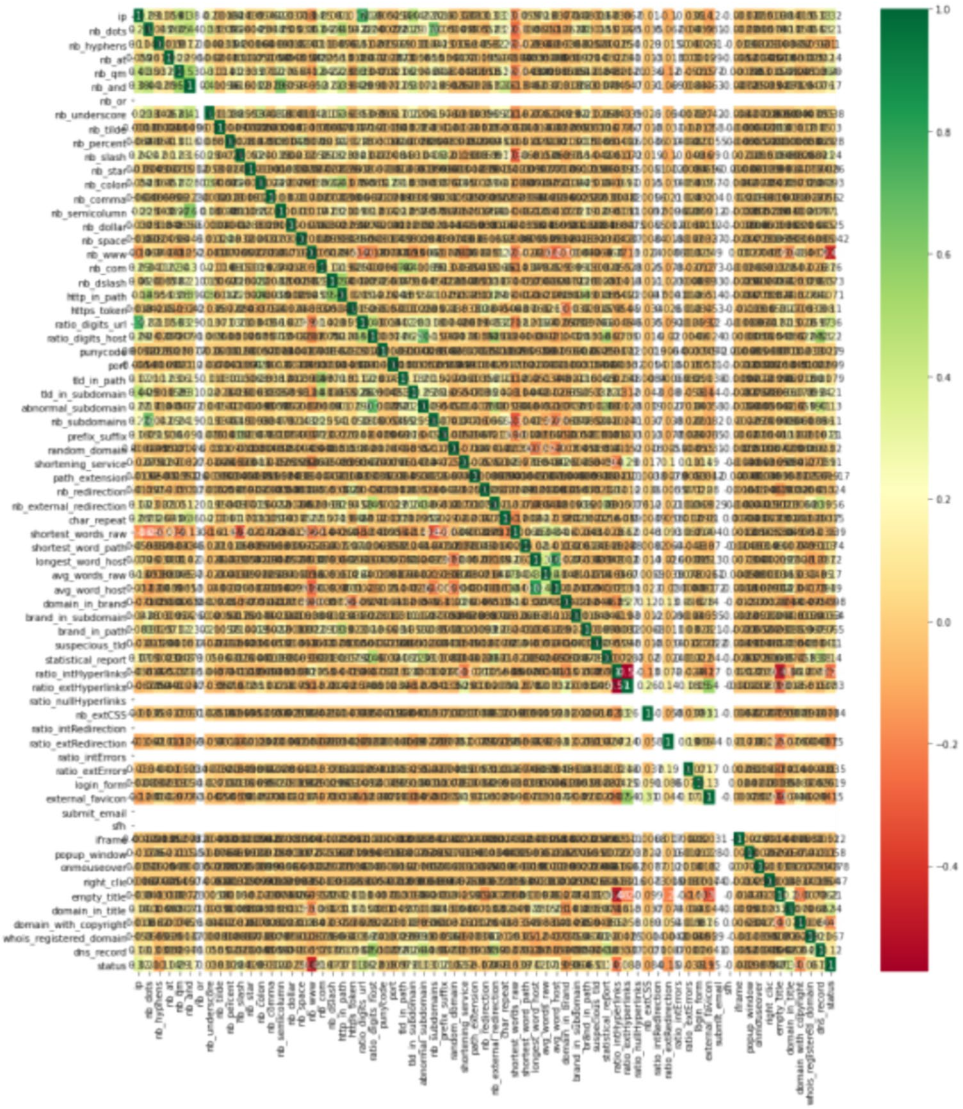


Fig. 5 Second phishing dataset by Pearson correlation

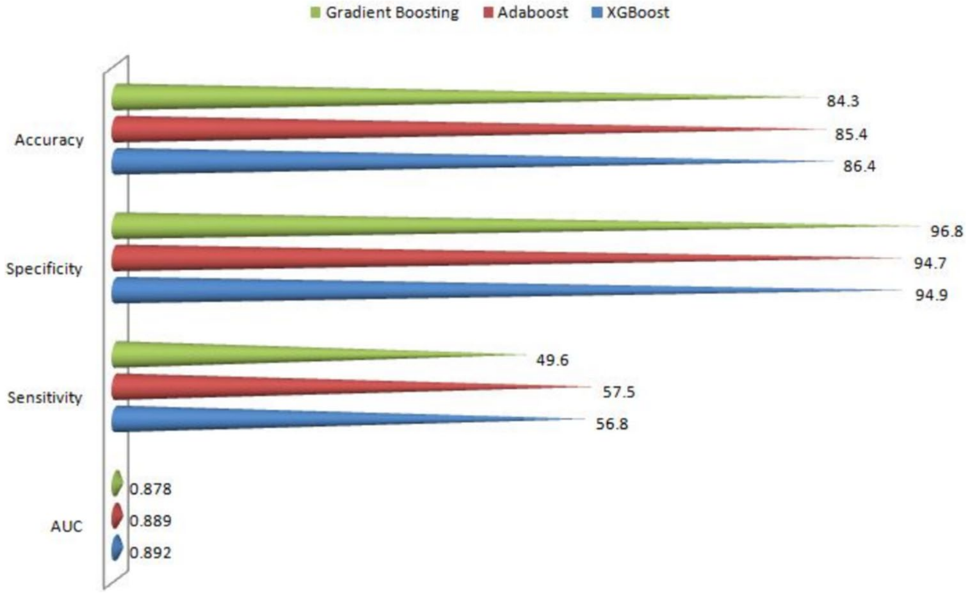


Fig. 6 Second phishing dataset by boosting algorithms

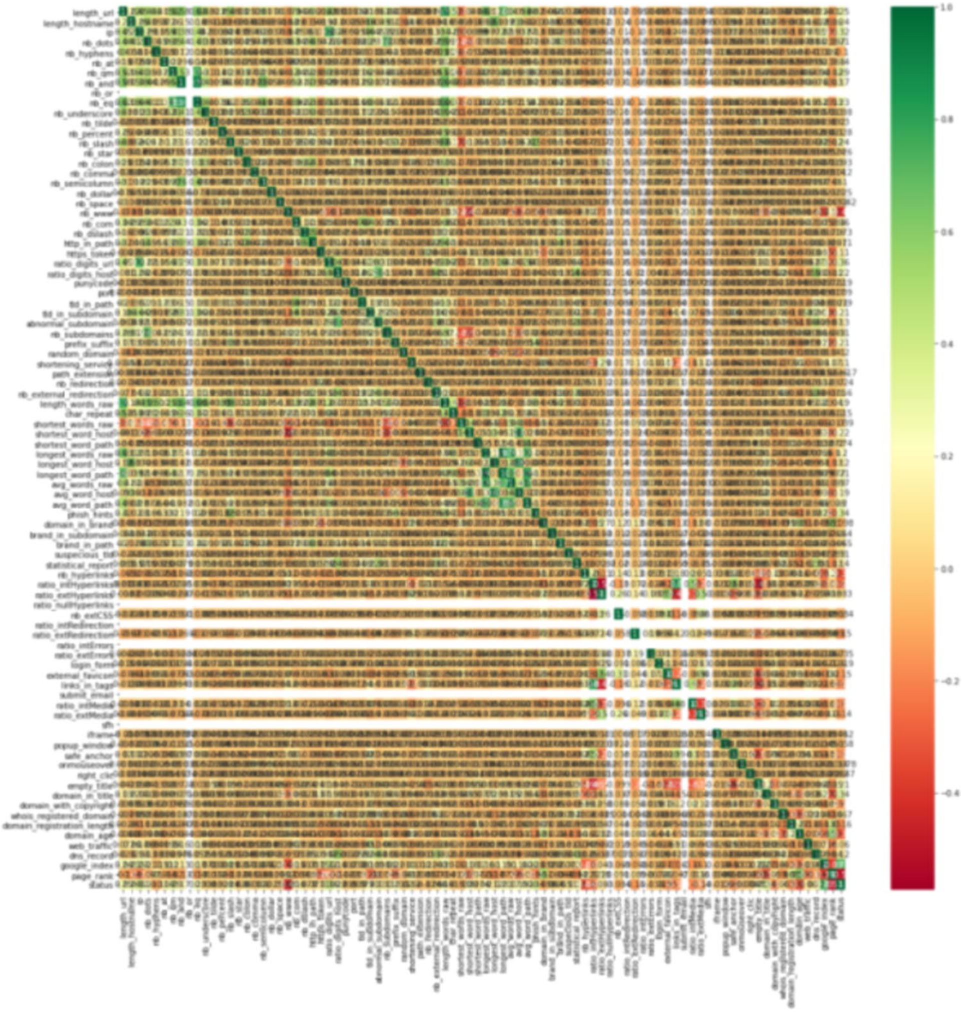


Fig. 7 Third phishing dataset by Pearson correlation

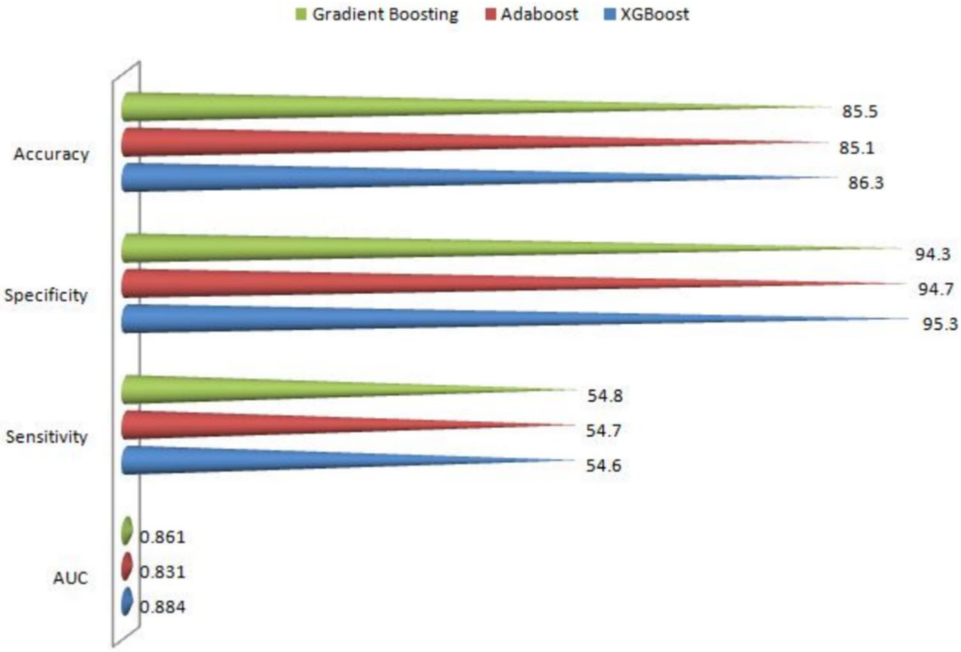


Fig. 8 Third phishing dataset by Boosting algorithms

Table 1 Accuracy results on phishing dataset by boosting algorithms

Models	Accuracy	Accuracy	Accuracy
XGBoost	87.5	86.4	86.3
Adaboost	86.4	85.4	85.1
Gradient boosting	85.6	84.3	85.5

4 Results

4.1 Experiment 1

Figure 3 used Pearson's correlation coefficients to analyse the relationships between phishing features. We multiply the variables for the sum and divide the sum to get the correlation coefficient for the findings. To demonstrate if a variable has a high or low correlation with another, a correlation matrix may be displayed (Saidi et al. 2019).

Models for the machine learning classifiers XGBoost (0.909, 61.7, 94.9, 87.5), gradient boosting (0.899, 57.3, 96.0, 86.4), and Adaboost (0.880, 62.2, 92.3, 85.6) are shown in Fig. 4. The analysis for the first organised phishing dataset using classifiers is shown in Fig. 3. According to the findings, XGBoost classifiers had the best AUC, sensitivity, and accuracy—85.6%, 62.2%, and 0.909%, respectively.

4.2 Experiment 2

The second phishing dataset is shown by Pearson correlation in Fig. 5. The correlation coefficient is calculated using the X and Y axis variables. The significance of correlation and Pearson's correlation coefficients were examined in Fig. 5 in regard to high and low correlations with one another (Chaurasia and Pal 2020; Yadav and Pal 2020).

Figure 6 shows the models for the machine learning classifiers XGBoost (0.892, 56.8, 94.9, 86.4), gradient boosting (0.889, 57.5, 94.7, 85.4), and Adaboost (0.878, 49.6, 96.8, 84.3.) in terms of AUC, sensitivity, specificity, and accuracy. The first organised phishing dataset employing classifiers is analysed in Fig. 6. The findings showed that XGBoost classifiers had the best accuracy, with an AUC of 0.892% and an accuracy rate of 86.4%, respectively.

4.3 Experiment 3

The third organised phishing dataset by Pearson correlation is shown in Fig. 7. The correlation coefficient is assessed by multiplying, dividing, and determining if there is a high or low correlation between the variables. Pearson's correlation coefficients and the significance of correlation were examined in Fig. 7 (Chaurasia and Pal 2014, 2022).

Figure 8 shows the models for the machine learning classifiers XGBoost (0.884, 54.6, 95.3, 86.3), gradient boosting (0.861, 54.8, 94.3, 85.5), and Adaboost (0.831, 54.7, 94.7, 85.1) in terms of AUC, sensitivity, specificity, and accuracy. The first organised phishing dataset employing classifiers is analysed in Fig. 8. The findings showed that XGBoost classifiers had the best accuracy, with an AUC of 0.884% and an accuracy of 86.3%, respectively.

5 Discussion

After the assessment trial, we had a conversation about better predictors (Table 1). The classification accuracy of XGBoost (87.5, 86.4, 86.3), Adaboost (86.4, 85.4, 85.1), and gradient boosting (85.6, 84.3, 85.5) was calculated by using boosting techniques to assess accuracy findings on phishing datasets. When compared to other classifiers, we found that the XGBoost classifier provided a higher calculated accuracy (87.5) in experiment 1, and it also produced a high calculated accuracy in experiments 2 and 3. While experiment 3 used a hybrid dataset from the prior two experiments, experiments 1 and 2 evaluated the data using two separate feature datasets. Ultimately, we found that for phishing datasets, the XGBoost classifier performs better in training and testing than gradient boosting and Adaboost. This study uses supervised machine learning (both regular and phishing). Furthermore, supervised machine learning produces excellent outcomes by reducing mistakes.

6 Conclusion

Three normalised organising complicated phishing datasets were used in this investigation. Through experimentation, we trained the first phishing dataset using gradient boosting, AdaBoost, and XGBoost, and then prepared the results. Using these three distinct machine learning techniques, train a second, distinct phishing dataset in the second manner. In the end, we merged the phishing datasets 1 and 2, and then we tested the prediction on the combined dataset 3. After making all of the predictions, we discovered that XGBoost outperforms the machine learning methods AdaBoost and gradient boosting. We intend to build on this in the future by predicting user-beneficial outcomes using real online datasets and a variety of ensemble models.

Data availability The datasets generated and/or analysed during the current study are available in the UCI Machine Learning Repository,https://archive.ics.uci.edu/ml/index.php.

Declarations

Conflict of interest The authors declare no competing interests.

References

Ahmad SBS, Rafie M, Ghorabie SM (2021) Spam detection on Twitter using a support vector machine and users' features by identifying their interactions. Multimed Tools Appl 80(8):11583–11605

Aiyar S, Shetty NP (2018) N-gram assisted Youtube spam comment detection. Procedia Comput Sci 132:174–182

Alauthman M (2020) Botnet spam e-mail detection using deep recurrent neural network. Int J 8(5):1979–1986

Alharthi R, Alhothali A, Moria K (2021) A real-time deep-learning approach for filtering Arabic low-quality content and accounts on Twitter. Inf Syst 99:101740

Alkawaz MH, Steven SJ, Hajamydeen AI (2020) Detecting phishing website using machine learning. In: 2020 16th IEEE International Colloquium on Signal Processing & Its Applications (CSPA). IEEE, pp 111–114

Bikmukhametov T, Jäschke J (2019) Oil production monitoring using gradient boosting machine learning algorithm. IFAC-Papersonline 52(1):514–519

Chaurasia V, Pal S (2014) Performance analysis of data mining algorithms for diagnosis and prediction of heart and breast cancer disease. Rev Res 3(8):1–13

Chaurasia V, Pal S (2020) Applications of machine learning techniques to predict diagnostic breast cancer. SN Comput Sci 1(5):1–11

Chaurasia V, Pal S (2022) An ensemble framework-stacking and feature selection technique for detection of breast cancer. Int J Med Eng Inform 14(3):240–251

Chen W, Lei X, Chakrabortty R, Pal SC, Sahana M, Janizadeh S (2021) Evaluation of different boosting ensemble machine learning models and novel deep learning and boosting framework for head-cut gully erosion susceptibility. J Environ Manage 284:112015

Chiew KL, Yong KSC, Tan CL (2018) A survey of phishing attacks: their types, vectors and technical approaches. Expert Syst Appl 106:1–20

Chu Z, Widjaja I, Wang H (2012) Detecting social spam campaigns on twitter. In: International Conference on Applied Cryptography and Network Security . Springer, Berlin, Heidelberg, pp. 455–472

Curtis SR, Rajivan P, Jones DN, Gonzalez C (2018) Phishing attempts among the dark triad: patterns of attack and vulnerability. Comput Hum Behav 87:174–182

Inuwa-Dutse I, Liptrott M, Korkontzelos I (2018) Detection of spam-posting accounts on Twitter. Neurocomputing 315:496–511

Jain A, Gairola R, Jain S, Arora A (2018) Thwarting spam on facebook: identifying spam posts using machine learning techniques. Available at https://arxiv.org/abs/1703.09398. Accessed 8 Jan 2023

Liu Y, Pang B, Wang X (2019) Opinion spam detection by incorporating multimodal embedded representation into a probabilistic review graph. Neurocomputing 366:276–283

Méndez JR, Iglesias EL, Fdez-Riverola F, Díaz F, Corchado JM (2005) Tokenising, stemming and stopword removal on anti-spam filtering domain. In: Conference of the Spanish Association for Artificial Intelligence . Springer, Berlin, Heidelberg, pp 449–458

Ruskanda FZ (2019) Study on the effect of preprocessing methods for spam email detection. Indones J Comput (Indo-JC) 4(1):109–118

Saidani N, Adi K, Allili MS (2020) A semantic-based classification approach for an enhanced spam detection. Comput Secur 94:101716

Saidi R, Bouaguel W, Essoussi N (2019) Hybrid feature selection method based on the genetic algorithm and pearson correlation coefficient. In: Hassanien A (eds) Machine Learning Paradigms: Theory and Application. Studies in Computational Intelligence, vol 801. Springer, Cham. https://doi.org/10.1007/978-3-030-02357-7_1

Solomatine DP, Shrestha DL (2004) AdaBoost. RT: a boosting algorithm for regression problems. In: 2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No. 04CH37541), Vol. 2. IEEE, pp 1163–1168

Yadav DC, Pal S (2020) Prediction of thyroid disease using decision tree ensemble method. Human Intell Syst Integr 2(1):89–95

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.