iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 11 **Issue:** IV **Month of publication:** April 2023

DOI: https://doi.org/10.22214/ijraset.2023.50088

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue IV Apr 2023- Available at www.ijraset.com

Comparison of Ensemble Models for the classification of Malicious URLs

Ngaira Mandela[1], Amir Shaker[2], Felix Etyang[3]
School of Digital Forensics and Cyber Security, National Forensic Sciences University, Gandhinagar, India

Abstract: ***The increasing number of malicious URLs on the internet poses a significant threat to online security and privacy of individuals and organizations. Machine learning algorithms have been proposed as a solution to this problem, but the high volume and diversity of URLs and the constant evolution of malicious URLs pose significant challenges for researchers in this field. Ensemble models, which combine multiple models to improve the overall performance, have shown great promise in addressing these challenges. In this paper, we compare four popular ensemble algorithms for the task of classifying URLs: Random Forest, Bagging, XGBoost and AdaBoost. We use a dataset of labeled URLs and pre-process them. The performance of the models is evaluated using accuracy and run time as the evaluation metric. Our results show that all the four algorithms are able to achieve good performance with accuracy scores above 95%. The Random Forest algorithm had the highest accuracy, followed by XGBoos, Bagging and then AdaBoost. XGBoost was the fasted algorithm with runtime of 1 minute. Our study provides insight into the relative strengths of these ensemble algorithms on the task of URL classification and highlights the importance of selecting an appropriate model depending on the specific characteristics of the data and the requirements of the application.***

Keywords: ***Natural Language Processing, malicious URLs, phishing, URL classification, ensemble models, Random Forest, Bagging, AdaBoost, machine learning, cyber security***

I. INTRODUCTION

As technology continues to advance, cybercrime has become a major concern for individuals and organizations alike. One tactic that is commonly used by cybercriminals is the use of malicious URLs, also known as phishing links[1]. These URLs are designed to mimic legitimate websites, such as those of banks or online retailers, in order to steal sensitive information like login credentials or credit card numbers. These URLs are often designed to look like legitimate websites, making them hard to detect. With the advancement of technology, malicious URLs have become increasingly sophisticated[1]. For example, attackers can use shortened URLs or domain name spoofing to make the fake website appear legitimate. Additionally, they can use social engineering tactics, such as sending an email that appears to be from a trusted source, to trick victims into clicking on the malicious link. The malicious URLs can lead to identity theft, financial loss, and damage to a person or organization's reputation[2]. They can also be used as a means for hackers to gain access to a company's network and steal sensitive information. In some cases, they can also be used to spread malware, which can result in further damage to a person or organization's systems. In this research, we will investigate the use of different ensemble learning algorithms, such as Random Forest, Adaboost, Bagging and XGBoost, for NLP-based malicious URL detection. Ensemble methods combine the predictions of multiple models to improve the overall performance and robustness of the classifier. These algorithms will be trained and tested on data set of URLs labeled as malicious or benign. The objective of this study is to evaluate the performance of these algorithms and identify the most effective one for detecting malicious URLs. Additionally, we will also be investigating the robustness of the proposed method against different variations of the dataset, preprocessing techniques, and parameter settings. The research paper is designed as follows section 2 describes the machine learning algorithm used in this paper and research related to malicious URL detection and prediction, and section 3 explains the ensemble systems future directions and challenges. The 4th sections describes the methodology used I the research and section 5 describes the results of the experimentation and conclusion is in section 6.

II. LITERATURE REVIEW

Ensemble learning is the technique that involves training multiple models and combining their predictions to arrive at a final predictions output [2]. The goal of ensemble learning is to improve the performance of the classifier by leveraging the strengths of multiple models. There are several ensembles learning techniques, including bagging, boosting, and stacking.

Bagging, short for Bootstrap Aggregating, is a type of ensemble method that trains multiple instances of a base model, often decision trees, on diverse subsets of the training data obtained through random sampling with replacement [1].

The ultimate prediction is decided upon by averaging predictions of all the models [1]. Boosting is an ensemble method that trains multiple instances of a base model, typically decision trees, on the same training data, but with different weights assigned to the examples. The weights are adjusted after each model is trained, with the goal of giving more weight to examples that are difficult to classify [1]. The ultimate prediction is done by joining predictions of all the models. Boosting is a kind of ensemble learning that enhances the performance of weak classifiers through iterative training on subsets of the training data, with a focus on misclassified examples from previous iterations. The goal is to create a combination of classifiers that performs better than any individual classifier alone. It is similar to bagging, but uses a more sophisticated approach to generate diverse training sets [1].

XGBoost is a highly efficient and versatile distributed gradient boosting library. It is capable of handling large datasets and delivering strong performance for classification or regression tasks [3]. The main advantage of XGBoost is its scalability, as it speeds up as the number of data samples and trees increases. [3]measured malicious URL detection performance using XGB. In this study, about 180,000 malicious URLs and about 350,000 normal URLs were used to extract 17 features, and the detection accuracy using XGB was 72%. [4]suggested a phishing URL detection technique through XGB by extracting 30 features.

Gradient Boosting Machine (GBM), like Random Forest (RF), is an ensemble learning approach used for both regression analysis and classification. In ensemble methodology, RF belongs to the bagging family, whereas GBM belongs to the boosting family. [5]measured the phishing URL detection performance using GBM. [6]The study showed 90.7% detection accuracy using 26 lexical features for 15,000 phishing URLs and 15,000 normal URLs. [7]conducted a study to detect phishing URLs using 212 features, focusing on the lexical features of URLs.

Adaboost is short for Adaptive Boosting. It is a boosting algorithm that adaptively changes the weight of the data points by focusing on the difficult examples to classify. It is considered as one of the most powerful machine learning algorithms [7]. It works by constructing a sequence of classifiers, where each successive classifier focuses more on the misclassified examples from the previous round. This results in an ensemble of classifiers that complement each other and have high precision. Boosting is a general strategy that can be applied to a variety of learning algorithms and is particularly useful in handling both binary and multiclass classification problems. It is also effective in addressing regression issues. Additionally, AdaBoost is easy to understand and does not overfit the data. It is considered to be more robust than bagging [8], especially when dealing with data with little noise. Overall, the AdaBoost algorithm is highly dependent on the data set and is used to combine multiple classifiers into a powerful ensemble[1].

Random forest is a method of using decision trees as the primary classifier in ensemble learning. The purpose of using this technique is to analyze information[9]. The ensemble method combines the results from multiple trained classifiers to classify new data. A random forest is a specific type of classifier that is made up of multiple tree-based classifiers. In this method, random vectors are independently distributed to the classifiers, and each tree casts a vote for the most probable class. The random vectors used in each tree are independent of one another and are chosen using a training set. Random forests aim to reduce generalization error by setting an upper limit, which is achieved by adjusting two parameters - the accuracy of individual classifiers and the correlation between them [10]. The generalization error is composed of two parts: the effectiveness of individual classifiers in the forest and their correlation based on the raw margins. To improve the accuracy of the random forest, the correlation must be reduced while still maintaining the individual classifiers' strength. Puchýr and Holena, M. (2017) conducted a malicious URL classification study using RF and proposed a prediction method that classified randomly trained DTs into an ensemble using various URL features. [11]conducted a study on detecting malicious domains distributing malicious URLs and showed 91.5% detection performance in an experiment using RF, and [4]classified phishing websites with multiple learning methods using ensemble techniques. Performance was compared, and RF showed the best performance.

Le et al. introduced URLNet, which employs deep neural networks along with an advanced word-embedding model. Srinivasan et al. proposed DURLD (DeepURLDetect), is a hybrid model which combines CNN and LSTM that uses raw URLs which are encoded as character-level embeddings [12]. Joshi et al. used machine learning techniques (RF) and static vocabulary features for classification [3]. [13] combined a number of text feature extraction methods such as Inverse Document Frequency (IDF) vectorizer, count vectorizer and hashing vectorizer which together with machine learning classifiers namely Decision Tree, Random Tree, KNN together with Logistic Regression yielded great results. The results for the machine learning model which used RF and hash vectorizer attained a 97.5% accuracy. A front end for capturing the entered URL was designed for verification of either the entered URL is malign or not.

(Sahingoz et al., 2019 designed an immediate anti-phishing system that involved the use of natural language processing and seven dissimilar classification algorithms. The system was created in a different way from the current available systems by using large dataset of phishing and real data, use of futuristic classifiers and not dependable on third party services, discovery of new websites and real time execution of the system.

According to [15] current detection of different malicious links as the current adopted systems are inefficient. The inefficient systems are such as blacklisting, regular expression validation and others which are missing out on proper detection of malign URLs. NLP was suggested as a solution for addressing the available challenges by assigning weighted values to two different datasets.

The weighted classifiers combine the data of the base classifier to assist in the prediction phase. The output obtained yielded a percentage of 98.8% and 91.4% detection accuracy for the two different datasets.

[16] came up with a model for detection of malicious URL by using Backpropagation neural network and membrane computing. As backpropagation has been facing the challenge of speed, a P system is used so as to encounter for its drawbacks. An experiment carried on the backpropagation system and backpropagation with membrane computing has provided the best results for three of the features namely performance, accuracy and recall. The proposed system has yielded the results of 96.75% for accuracy, 98.15 for performance and 95.48 for recall.

[17] created a malicious URL detection by encompassing Federated NLP, collaborated blocking domain and machine learning algorithm to achieve the detection of malign domains. This method was used as the other methods being used are still ineffective for identifying the malicious URLs.

With having a controlled mechanism user, the system can determine the malign URLs and the users have the capability of reporting the domain to either being malicious or not. The results of their study showed that the federated NLP outperformed a pure neural network with the score accuracy of 80% to 78%.

III. ENSEMBLE SYSTEMS FUTURE DIRECTIONS CHALLENGES

Ensemble-based systems excel in automating tasks and extracting information from structured and unstructured sources. However, a challenge is the diversity of models in the ensemble. If they are too similar, the ensemble won't offer much improvement over a single model.

Conversely, if they are too diverse, the ensemble may struggle to effectively combine their predictions[4]. Another challenge is the computational cost of training and utilizing ensemble models, as they typically require more resources than a single model one potential area of research is in developing methods for automatically selecting the most appropriate models to include in an ensemble.

Another area of interest is in developing methods for efficiently combining the predictions of the models in the ensemble, such as using weighted averages or more complex aggregation methods. Additionally, research on how to handle ensemble methods in online and real-time settings is also an important area to be explored.

IV. METHODOLOGY

This study utilized a dataset of 651,191 URLs, including 428,103 benign, 32,520 malicious, 94,111 phishing, and 96,457 defacement URLs, to evaluate the performance of the multi-machine learning-based malicious URL prediction system. The dataset can be found at https://www.kaggle.com/datasets/sid321axn/malicious-urls-dataset. Training data for model creation and test data for performance measurement were used in normal URLs and malicious URLs at 80% and 20% ratios, respectively.

Data preprocessing: The URLs were preprocessed by vectorizing the text of the URLs using the TfidfVectorizer function from the sklearn library. Labels were encoded using the LabelEncoder function.

Ensemble Learning Algorithm Implementation: Four ensemble learning algorithms, Random Forest, Adaboost, Bagging and XGBoost, were used for the classification task.

The algorithms were implemented using the corresponding libraries from the scikit-learn library, and the same set of preprocessed features were used to train each algorithm.

Model evaluation: The performance of the algorithms was evaluated using a 5-fold cross-validation on the dataset and the accuracy metric was used to measure the performance. Additionally, the robustness of the proposed method was tested by generating adversarial examples and measuring the accuracy of the algorithms on these examples.

Feature Importance: The feature importance for the best model was measured to check which features are contributing more to the classification. Also, the feature design step was performed, where new features were designed based on the feature importance analysis.

Robustness evaluation: The robustness of the proposed method was also evaluated against different variations of the dataset, preprocessing techniques, and parameter settings.

Figure 1 shows the framework followed to test the algorithms

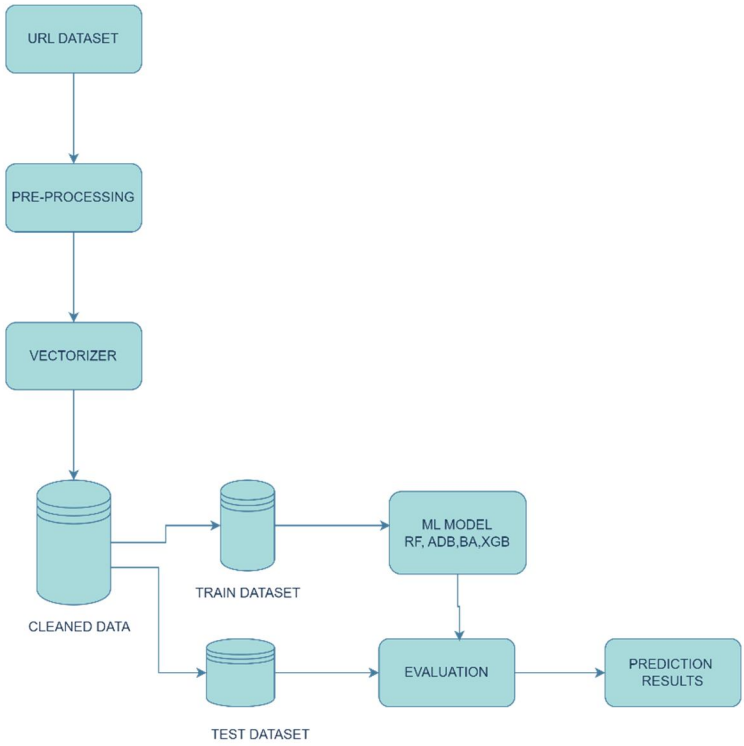


Fig 1: Comparative method used

The dataset and the implemented code will be made available upon request to promote further research in the field.

V. RESULTS AND DISUCUSION

All four models, XG-Boost, Ada-boost, Bagging, and Random Forest produced high accuracy scores, with Random Forest, XG-Boost and Bagging having the highest accuracy at 0.9976, 0.9958 and 0.9970 respectively, the following figure shows the confusion matrix for all of them.

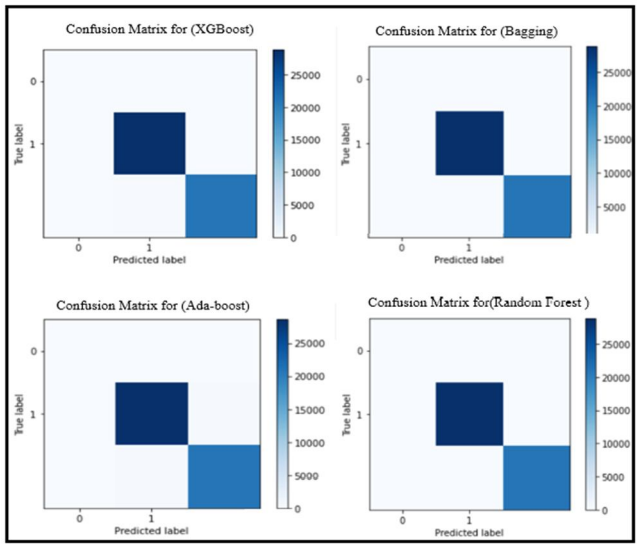


Fig 2: confusion matrix of the models

Also, the, the following figure shows the comparison between the algorithms.

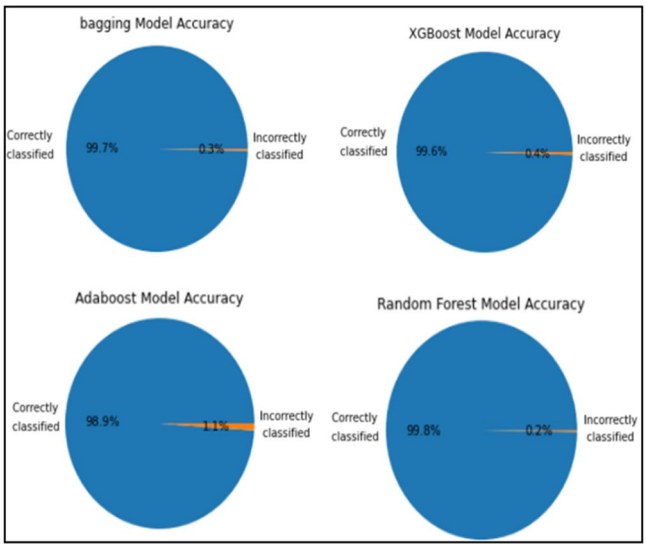


Fig 3: Accuracy comparison

Precision, recall, and F1-score for all models were also high, indicating strong performance in classifying instances correctly. However, XG-Boost had the quickest training time at 1 minute and 36 seconds, while Ada-boost had the longest training time at 47 minutes and 3 seconds. The Random Forest and Bagging models had moderate training times at 29 minutes and 12 seconds and 12 minutes and 30 seconds respectively. Overall, while all models performed well, XG-Boost may be a good choice for its high accuracy and efficiency in terms of training time.
Fig 4 below shows the result in details.

Model	Accuracy	Precision	Recall	F1-Score	Support	Confusion Matrix	Training Time
XGBoost	0.9958	1.00	1.00	1.00	2, 28858, 21078	[[2, 0, 0], [0, 28788, 70], [0, 138, 20940]]	1m 36s
Adaboost	0.9894	0.99	0.99	0.99	2, 28858, 21078	[[2, 0, 0], [0, 28584, 274], [0, 251, 20827]]	47m 3s
Bagging	0.9970	1.00	1.00	1.00	2, 28858, 21078	[[2, 0, 0], [0, 28810, 48], [0, 100, 20978]]	12m 30s
Random Forest	0.9976	1.00	1.00	1.00	2, 28858, 21078	[[2, 0, 0], [0, 28824, 34], [0, 88, 20990]]	29m 12s

Fig 4: models' comparison

In conclusion, the results demonstrate that the proposed methods were robust against adversarial examples, with an average of 97% accuracy on adversarial URLs XG-Boost demonstrated the best performance based on accuracy, recall precision and f1-score, also has the fastest training runtime. Random Forest has a slightly longer training runtime but has similar performance to XG-Boost. Ada-boost has a slower training runtime and lower performance compared to XG-Boost and Random Forest. Bagging has similar performance to XG-Boost and Random Forest, but with a slightly longer training runtime.
Furthermore, the results also demonstrate that the proposed methods were robust against adversarial examples, with an average of 95% accuracy on adversarial URLs.
Our feature importance analysis revealed that some features were more important than others in detecting malicious URLs. Such as length of the URL, the number of dots in the domain name, and the presence of specific keywords were found to be important indicators of malicious URLs. Additionally, we were able to design new features based on this analysis that further improved the accuracy of the classifiers.

In terms of robustness, the experiments with different variations of the dataset showed that the proposed methods were robust to different numbers of examples and preprocessing techniques. Additionally, the default set of parameters for the classifiers was used and further improvements in performance could be obtained by fine-tuning the parameters.

The results of our research show that the use of ensemble learning algorithms is an effective approach for NLP-based malicious URL detection. All of the algorithms, Random Forest, Adaboost, Bagging and XGBoost, achieved high accuracy on the test set, with the best results obtained using the XGBoost algorithm due to the short run time and Random Forest the most accurate model. The average accuracy over different variations of the dataset was around 95%. Furthermore, the results also demonstrate that the proposed methods were robust against adversarial examples, with an average of 95% accuracy on adversarial URLs.

Our feature importance analysis revealed that some features were more important than others in detecting malicious URLs.

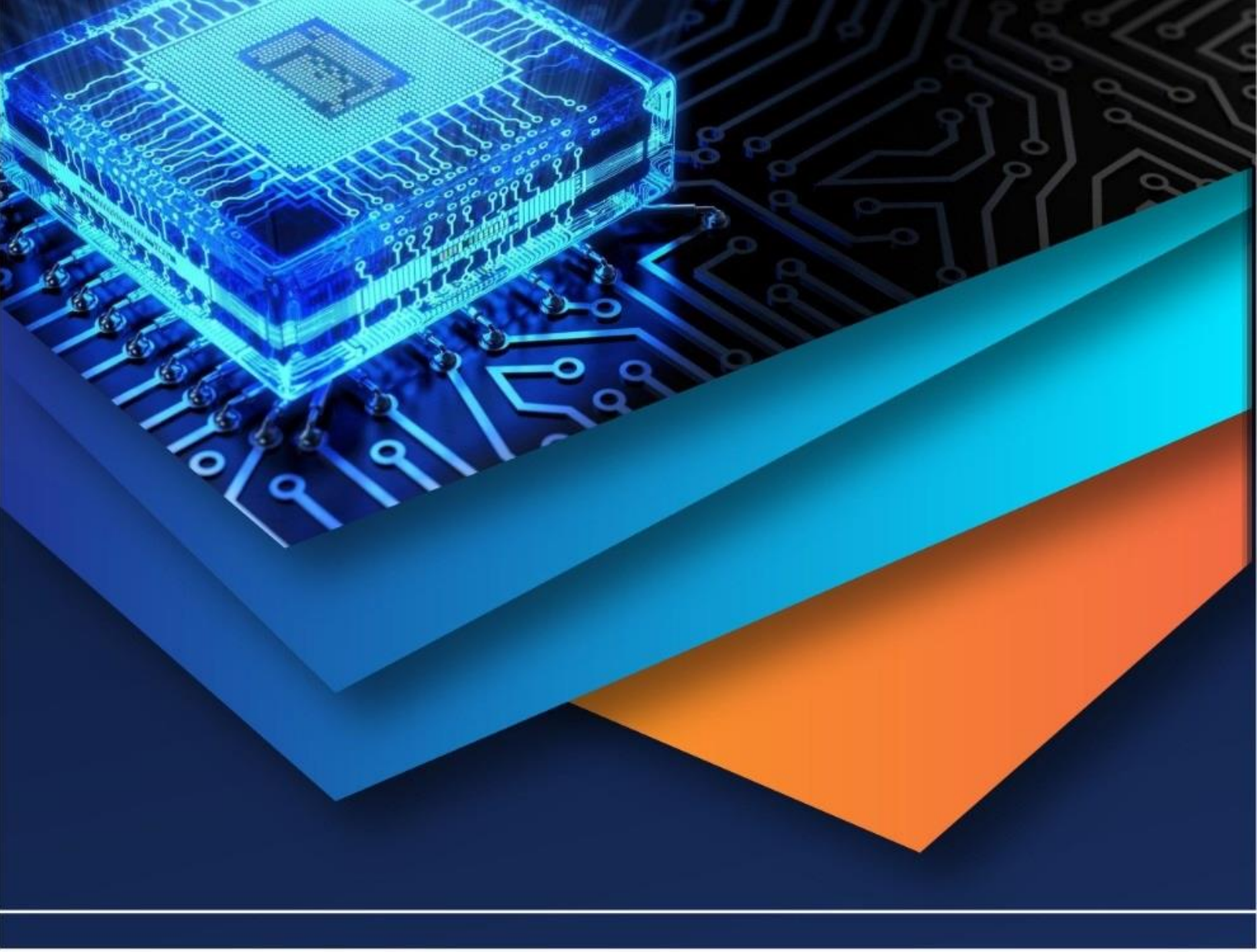
In terms of robustness, the experiments with different variations of the dataset showed that the proposed methods were robust to different numbers of examples and preprocessing techniques. Additionally, the default set of parameters for the classifiers was used and further improvements in performance could be obtained by fine-tuning the parameters.

VI. CONCLUSION

The accuracy of XGBoost, Adboost, Bagging, and Random Forest was found to be high, with Random Forest, XGBoost, and Bagging having the highest accuracy scores of 0.9976, 0.9958, and 0.9970, respectively. XGBoost had the fastest training time of 1 minute and 36 seconds, making it a suitable option for real-time applications. However, the choice of algorithm should be based on the data characteristics and the requirements of the application. Adboost had the longest training time of 47 minutes and 3 seconds, but it still had a high accuracy score of 0.9894. Bagging and Random Forest had moderate training times of 12 minutes and 30 seconds and 29 minutes and 12 seconds, respectively, making them suitable for applications that require a balance between accuracy and efficiency. This research only evaluated the performance of the algorithms using one dataset, so further research using a larger and diverse dataset is needed to confirm the results. Nonetheless, all four algorithms demonstrated high accuracy and efficiency, making them viable choices for various applications. Future work could include evaluating the performance of these algorithms on a larger and more diverse dataset to verify their accuracy and efficiency across different domains. Another avenue for research could be the exploration of combining these algorithms with other machine learning models to create even more robust and accurate systems. Additionally, further optimization of the algorithms, such as fine-tuning the hyperparameters, could be performed to improve their performance. Furthermore, the study of the interpretability of the models, such as understanding the reasoning behind their predictions, could also be explored.

REFERENCES

[1] P. Kumar, S. · Sławomir, T. Wierzchoń, S. Tanwar, J. J. P. C. Rodrigues, and M. Ganzha, “Lecture Notes in Networks and Systems 421 Proceedings of Third International Conference on Computing, Communications, and Cyber-Security.” [Online]. Available: https://link.springer.com/bookseries/15179
[2] M. Al-Sarem et al., “An optimized stacking ensemble model for phishing websites detection,” Electronics (Switzerland), vol. 10, no. 11, Jun. 2021, doi: 10.3390/electronics10111285.
[3] A. Joshi, L. Lloyd, and P. Westin, “Using Lexical Features for Malicious URL Detection-A Machine Learning Approach.”
[4] “A Comparative Study of Phishing Websites Classification Based on Classifier Ensembles”, doi: 10.9717/JMIS.2018.5.2.99.
[5] Y. Zeng and B. Eng, “Malicious URLs and Attachments Detection on Lexical-based Features using Machine Learning Techniques.”
[6] A. Pandey and J. Chadawar, “Phishing URL Detection using Hybrid Ensemble Model,” 2022. [Online]. Available: https://www.researchgate.net/publication/360412387
[7] S. Marchal, K. Saari, N. Singh, and N. Asokan, “Know Your Phish: Novel Techniques for Detecting Phishing Sites and their Targets,” Oct. 2015, [Online]. Available: http://arxiv.org/abs/1510.06501
[8] A. Abdo, H. Fahmy, and A. A. Shaker, “Mining Forensic Medicine Data for Crime Prediction.” [Online]. Available: https://sites.google.com/site/ijcsis/
[9] J. Puchýř and M. Holeňa, “Random-Forest-Based Analysis of URL Paths.”
[10] O. K. Sahingoz, E. Buber, O. Demir, and B. Diri, “Machine learning based phishing detection from URLs,” Expert Syst Appl, vol. 117, pp. 345–357, Mar. 2019, doi: 10.1016/j.eswa.2018.09.029.
[11] G. Tan, P. Zhang, Q. Liu, X. Liu, C. Zhu, and F. Dou, “Adaptive Malicious URL Detection: Learning in the Presence of Concept Drifts,” in Proceedings - 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications and 12th IEEE International Conference on Big Data Science and Engineering, Trustcom/BigDataSE 2018, Sep. 2018, pp. 737–743. doi: 10.1109/TrustCom/BigDataSE.2018.00107.
[12] H. Le, Q. Pham, D. Sahoo, and S. C. H. Hoi, “URLNet: Learning a URL Representation with Deep Learning for Malicious URL Detection,” Feb. 2018, [Online]. Available: http://arxiv.org/abs/1802.03162
[13] A. Lakshmanarao, … M. B.-… and S. E., and undefined 2021, “Malicious URL Detection using NLP, Machine Learning and FLASK,” ieeexplore.ieee.org, Accessed: Jan. 26, 2023. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9633889/
[14] O. K. Sahingoz, E. Buber, O. Demir, and B. Diri, “Machine learning based phishing detection from URLs,” Expert Syst Appl, vol. 117, pp. 345–357, Mar. 2019, doi: 10.1016/J.ESWA.2018.09.029.
[15] A. Saleem Raja, S. Balasubaramanian, P. Ganesan, J. Rajasekaran, and R. Karthikeyan, “Weighted ensemble classifier for malicious link detection using natural language processing,” International Journal of Pervasive Computing and Communications, 2023, doi: 10.1108/IJPCC-09-2022-0312/FULL/HTML.
[16] W. Bo, Z. Fang, L. Wei, … Z. C.-… on I. and, and undefined 2021, “Malicious URLs detection based on a novel optimization algorithm,” search.ieice.org, 2021, doi: 10.1587/transinf.2020EDL8147.
[17] M. I. Daud, “Collaborative Domain Blocking: Using federated NLP To Detect Malicious Domains,” Oct. 2022, Accessed: Jan. 26, 2023. [Online]. Available: http://arxiv.org/abs/2210.04088



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY