ISSN No:-2456-2165

Social Engineering Detection : Phishing URLs

Utkarsh Singh[1]
Dept. of CSE
Chandigarh University
Mohali, India

Ashvini Kumar[2]
Dept. of CSE
Chandigarh University
Mohali, India

Pratham Jain[3]
Dept. of CSE
Chandigarh University
Mohali, India

Tanya Jaiswal[4]
Dept. of CSE
Chandigarh University
Mohali, India

Sudhanshu Shekhar[5]
Dept. of CSE
Chandigarh University
Mohali, India

Gurleen Kaur[6]
Dept. of CSE
Chandigarh University
Mohali, India

Abstract:- In the digital age, the proliferation of malicious phishing URLs poses a significant threat to online security. While conventional machine learning algorithms have been employed to combat this menace, our research pioneers the use of ensemble methods, including XGBoost and Random Forest, for phishing URL detection. Our methodology involves collection of the data, preprocessing it then feature extraction followed by model training, evaluation and comparison. Notably, our results reveal the superior accuracy of ensemble methods in distinguishing phishing URLs from legitimate ones. These findings underscore the potential of ensemble methods as a game-changing asset in the battle against cyber threats, promising enhanced online security and the protection of sensitive user information.

Keywords:- *Social Engineering, Phishing URLs, Cyber Security, Machine Learning.*

I. INTRODUCTION

In the digital age, where the exchange of information and communication are paramount, individuals and organizations alike face an ever-increasing threat from social engineering attacks, with phishing being a notorious exemplar. Within this realm, one insidious tactic has emerged as a primary conduit for deceit and exploitation: phishing URLs. These malicious web links, often camouflaged as legitimate destinations, are designed to deceive unsuspecting users into divulging sensitive information or unleashing cyber threats.

Just like any file on a computer can be located by supplying its filename, any website can be located using a URL. Each Uniform Resource Locator (URL) has two primary components: the protocol and the resource identifier. The protocol is the first part of the URL, and it specifies the method used to access the resource. For example, HTTPS is a secure version of HTTP that is used to retrieve hypertext documents. Other protocols include File Transfer Protocol (FTP), Domain Name System (DNS), and more. The second part of the URL is the resource identifier, which is used to grant access to an online destination. For instance, in the URL https://www.google.com, the resource identifier is “www.google.com”.

Asadullah Safi [1] has described several types of phishing attacks, including email, web and link manipulation.

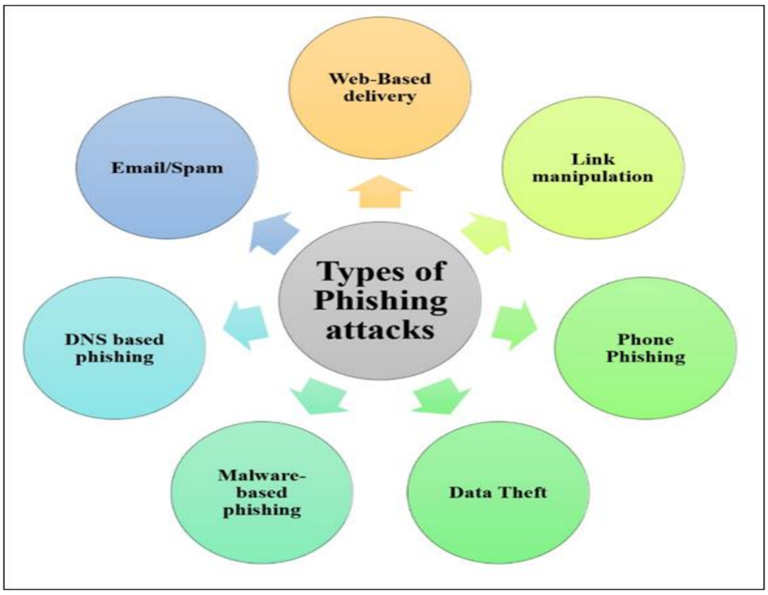

Fig 1 Types of Phishing Attacks

The requirement of robust and efficient mechanisms to detect phishing URLs has never been more critical. The stakes are high, encompassing not only the protection of personal data but also the preservation of trust in online transactions and communication.

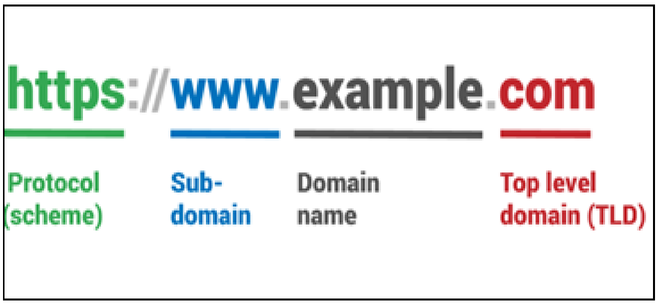

Fig 2 Example of a URL

This research paper delves into the domain of "Social Engineering Attack Detection: Phishing URLs." It focuses on harnessing the capabilities of multiple machine learning models, in combination with ensemble methods, to discern phishing URLs from their legitimate counterparts. This research strives to illuminate the efficacy of different models and their potential for enhancing the accuracy and timeliness of detection, ultimately bolstering cybersecurity defenses in a world where the preservation of digital trust is paramount.

II. RELATED WORK

The field of spam and social engineering detection has witnessed significant advancements over the years, with researchers proposing various techniques and models to combat these security threats. In this literature survey, we reviewed 6 papers on Phishing Detection Systems.

Qabajeh et al. [2] have recently devoted themselves to research on traditional and automatic phishing detection technology. Raising awareness, educating users, holding regular courses or seminars, and utilizing legal opinions are some of the strategies to prevent phishing. Product and machine learning techniques are discussed in the context of protection against computerized or automated phishing.

Kunju et al. [3] Use investigative methods to investigate phishing attacks. Research provides various techniques and solutions for detecting phishing attacks. Research shows that a number of proposed remediation measures are not sufficient to deal with phishing attacks.

Kathrine et al. [4] proposed a framework to detect and prevent various phishing attacks. This study proves that machine learning-based algorithms can identify real-world benefits. The literature examined in this project includes only 11 studies, and deep learning techniques used in combating phishing websites are not included in the studies. These are the limitations of this study.

Benavides et al. [5] conducted a review and analyzed different methods used by other researchers to use deep learning to detect phishing attacks. In summary, there are still large differences in deep learning algorithms for detecting phishing attacks. This study has only 19 articles published between 2014 and 2019 in the existing literature.

Arshad et al. [6] show different types of phishing and anti-phishing in their work. According to SLR's analysis, the most commonly used phishing tactics include spear phishing, email spoofing, phone phishing and email manipulation. The study found that machine learning methods were the most accurate.

Shantanu et al. [7] In his paper, decided to find bad URLs as a binary classification problem and evaluated the performance of several well-known machine learning classifiers. The model was trained using Kaggle's public database of 450,000 URLs.

Table 1 shows the details of data analysis of phishing detection systems.

Table 1 Phishing Detection Systems

Author and Year	Aim	Main Findings	Limitations
Qabajeh et al. [2], 2018	This review article contrasts conventional anti-phishing techniques, such as utilizing a legal viewpoint, educating users, holding recurring training sessions, and increasing awareness.	Machine learning and rule generation are ideal for stopping phishing attempts because of the high detection rate and, more importantly, the results are easy to understand.	Sixty-seven studies were evaluated, but the studies did not include an in-depth study.
Kunju et al. [3], 2019	This article provides an overview of various machine learning algorithms such as kNN, Naive Bayes, Decision Trees, SVM, Neural Networks and Random Forests to detect phishing websites.	This study indicates that detecting phishing websites with a single method is insufficient.	In the literature reviewed in this study, only 14 studies discussed machine learning.
Kathrine et al. [4], 2019	This project introduces various phishing attacks and the latest protection techniques. This study provides a framework for identifying and avoiding phishing scams.	This study shows that machine learning-based algorithms can identify real-world benefits.	Just 11 studies were covered in the work, and Deep Learning methods for phishing website mitigation are not included in the research.

ISSN No:-2456-2165

Author and Year	Aim	Main Findings	Limitations
Benavides et al. [5], 2020	The purpose of this literature review is to evaluate various proposals from other researchers for using deep learning to identify phishing attacks.	This project only considers the search terms phishing and deep learning, including 19 studies.	In summary, there is still a huge gap in the field of deep learning algorithms for detecting phishing attacks.
Arshad et al. [6], 2021	This study discusses various phishing strategies and protection against phishing.	They came to the conclusion that email manipulation, phone phishing, spear phishing, and email spoofing were the most often used phishing strategies.	The research only draws from twenty studies.
Shantanu et al. [7], 2021	This study examines various classification models to determine which one has the best accuracy on a dataset of phishing URLs.	In this paper, they address the binary classification problem of malicious URL detection and evaluate the performance of various popular machine learning classifiers.	The models in this work were not constructed using ensemble methods.

III. METHODOLOGY

In this research, we present our methodology for the robust detection of malicious URLs, with a specific focus on machine learning models, feature engineering, and ensemble methods for classification. We embark on this journey through a systematic set of steps.

We begin with the pivotal phase of data collection. The dataset [8] is taken from www.kaggle.com which includes 507195 Unique URLs out of which 72% are Good URLs and 28% are the Malicious ones as shown in Table 2. Data preprocessing follows, an indispensable step to ensure the integrity of the dataset. The data is diligently cleaned to eliminate inconsistencies and noise. We also perform feature extraction, deriving significant attributes from the URLs, including domain, path, length, and the presence of special characters. These extracted features will be instrumental as input variables for our machine learning models.

Table 2 Dataset Details

Good URLs	Malicious URLs
72%	28%
3,65,180	1,42,015

To effectively train the model and test, the data is divided into two groups: training and testing. The training process will enable our model to learn from past data, and the light test will be evidence of evaluating the model.

The initial selection of machine learning models [7] was diverse and included many types of learning. Choose models such as support vector machine (SVM), nearest neighbor (KNN), decision trees, random forest, gradient boosting, and packing and boosting transport integration. These models represent a wide range of distribution strategies. After model selection, the next step is the training phase. The selected model is trained on the training data, a process that involves fine-tuning hyperparameters to improve its performance.

Discover the power of collaborative processes to increase the efficiency of distribution. This includes looking at methods like random forest integration, gradient boosting integration (like XGBoost), AdaBoost, and Stacking.

The core of our research is the comparative analysis. We delve into the performance of each model in-depth, with a focus on both traditional and ensemble methods. Through this analysis, we dive into the strengths and limitations of each model and evaluate their accuracy and robustness in distinguishing malicious from legitimate URLs.

The below flow diagram describes the flow of our model which involves, firstly the Pre-processing phase followed by the detection phase. The Pre-processing phase contains webpage feature generation, extraction and feature vectorization. The detection phase contains training set and testing set, feature model training and result analysis.

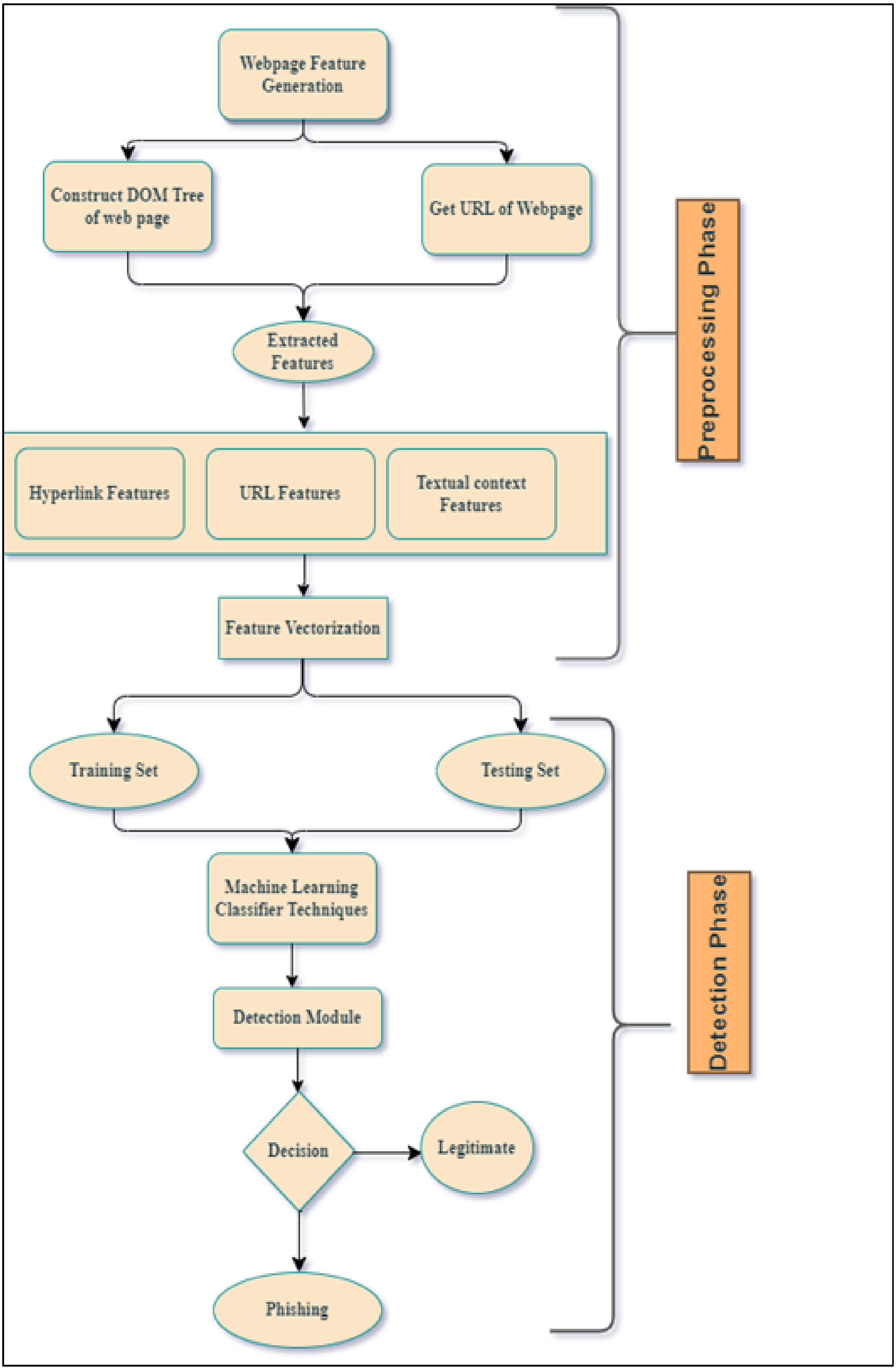

Fig 3 Phishing Model Flow Diagram

IV. EXPERIMENTAL ANALYSIS

Feature Extraction: Feature extraction [9] is the process of representing or enhancing features to make machine learning models more efficient. It helps in reducing the size and speeding up the work. The most common methods are discriminant analysis and principal component analysis.

Feature scaling: Feature scaling is a process of scaling data features within a fixed range. It is used during data preprocessing to handle high variance data. Without detailed information, machine learning models tend to give more weight to higher values and less weight to lower values. It is one of the most important and time-consuming steps in the previous document.

Large files are divided into 80-20 rules. Each model is trained on 80% of the data and tested on the remaining 20%.

➢ *Measurements used to Evaluate Classification Models:*

- True Positive (TP): Model predicts True and the result is also True.
- False Positive (FP): Model predicts True but the result is False
- True Negative (TN): Model predicts false and the result is also False.
- False Negative (FN): Model predicts False but the result is True.
- Accuracy: It is the true values divided by total number of values

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

Precision: The ratio of correct predictions to the total number of correct predictions.

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

Recall: It is predicted true values divided by the total actual true values.

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

F1-score: F score is the harmonic mean of precision and recall.

$$F1 - Score = 2 * \frac{Precision*Recall}{Precision+Recall} \quad (4)$$

Table 3 shows the position of TP, TN, FP and FN in a confusion matrix.

Table 3 Confusion Matrix

	Positive	**Negative**
Positive	True Positive	False Positive
Negative	False Negative	True Negative

Use the metrics above to train and evaluate different models. Two integration methods are used: Random Forest and XGBoost classifier. A prediction accuracy of 92.1% was achieved using random forest classification. A prediction accuracy of 93.7% was achieved using XGBoost.

➢ *Random Forest*

Random forest [10] is a popular machine learning algorithm suite that aims to reduce variance by using a series of deep decisions to train a model consisting of different domains of the same training; The results are then shown as average values to obtain the final classification.

The results of the random forest integrated model are shown in Figure 4. It shows the model's accuracy, precision, recall, and F-score.

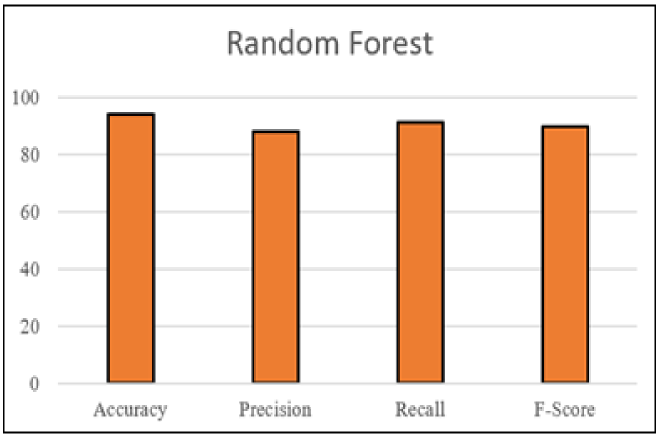

Fig 4 Random Forest Results

➢ *XGBoost*

XGBoost [11] is an efficient, adaptable, and portable gradient boosting algorithm. To get good results, it makes use of weighted classifiers, tree pruning, and parallelization.

The results of the XGBoost integrated model are shown in Figure 5. It shows the model's accuracy, precision, recall, and F-score.

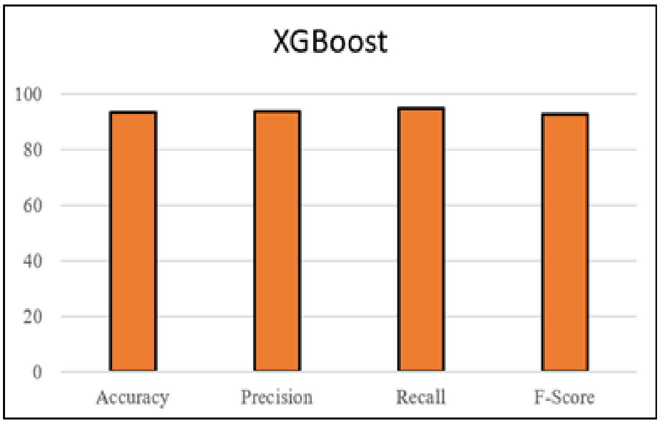

Fig 5 XGBoost Results

The confusion matrix values of random forest and XGBoost are shown in Table 4.

Table 4 Confusion matrix values

	Random Forest		XGBoost	
	Positive	**Negative**	**Positive**	**Negative**
Positive	242	32	220	18
Negative	22	235	12	180

The below Table 5 shows the summary of the test results of random forest and XGBoost.

Table 5 Summary of Test Results

Algorithm	Random Forest	XGBoost
Accuracy	0.921	0.937
Precision	0.883	0.938
Recall	0.914	0.949
F-Score	0.898	0.928

In Table 5, XGBoost accuracy, precision, recall and F-Score values are more than random forest.

V. COMPARATIVE ANALYSIS

Various classification models have been made earlier for classifying the phishing URLs into Safe or Malicious ones. One such work is done by Shantanu et. al. [7] where he chose non-ensembled training models Naïve Bayes, KNN and Support Vector Machines. Another one was Sharad Rajendra Parmar et. al. [12] who used algorithms Logistic Regression and KNN to train his model. Table 6 shows the comparative analysis of various algorithm results.

Table 6 Comparative Analysis of Various Algorithms

Author	Algorithm	Accuracy	Precision	Recall	F-Score
Shantanu et. al.	Naïve Bayes	0.891	0.881	0.843	0.876
	KNN	0.917	0.890	0.812	0.910
	SVM	0.921	0.901	0.842	0.913
Sharad et. al.	Logistic Regression	0.924	0.929	0.936	0.932
	KNN	0.543	0.605	0.548	0.756
Our Models	RF	0.921	0.883	0.914	0.898
	XGBoost	0.937	0.938	0.949	0.928

Below Fig. 6 Shows the Comparative Analysis of the algorithms used earlier and our ensemble methods.

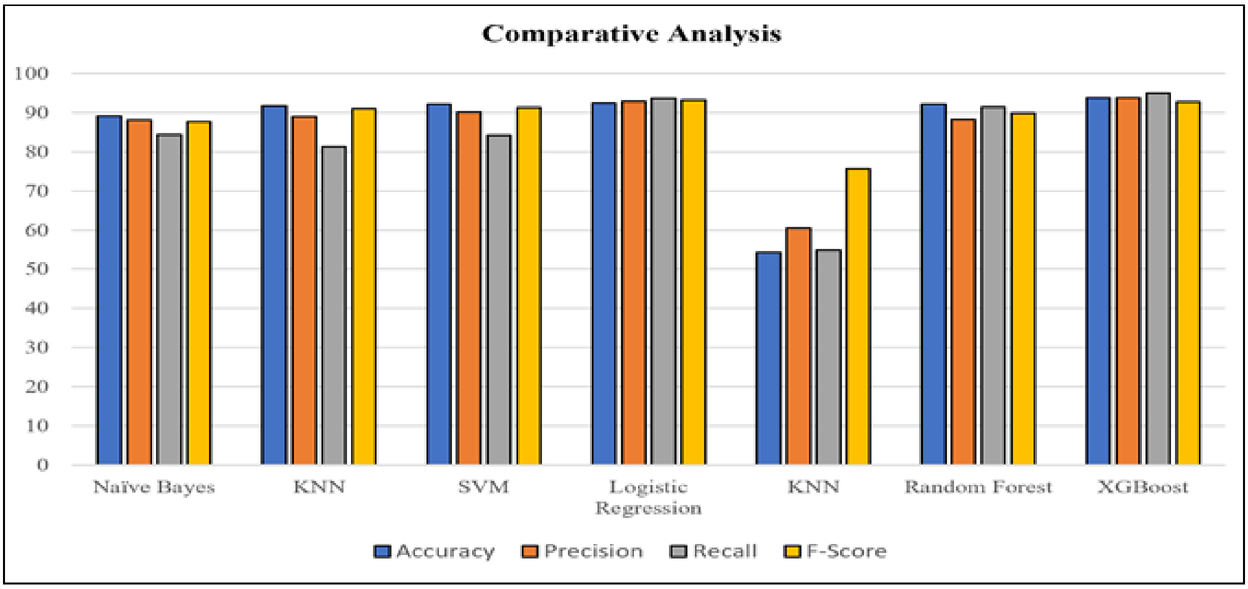

Fig 6 Comparative Analysis of Algorithms

From the above figure, we can see that our models – Random Forest and XGBoost have performed well in all the metrics like Accuracy, Precision, Recall and F-Score.

VI. CONCLUSION

To reduce phishing attacks or malware attacks, the learning process can be a very good technique because it can classify good and non-bad phishing URLs. All conditions are taken into account; We can say that learning together can produce good classification results. The rationale behind this is that ensemble learning solves a given problem by combining the best features of several models. This method significantly enhances the classification.

To get much better outcomes, other combinations of various machine learning models can be investigated in future studies. It is evident that the ensembled algorithms which are combinations give much better results than the individual machine learning algorithms.

REFERENCES

[1]. Asadullah Safi, Satwinder Singh, "A systematic literature review on phishing website detection techniques", Journal of King Saud University, Volume 35, Issue 2, 2023, pp. 590-611, ISSN 1319-1578

[2]. Qabajeh, I., Thabtah, F. 2018. "A recent review of conventional vs. automated cybersecurity anti-phishing techniques". Computer Sci. Rev. 29, 44– 55.

[3]. Kunju, M.V., Dainel, E., Anthony, H.C., Bhelwa, S., 2019. "Evaluation of phishing techniques based on machine learning", 2019 International Conference on Intelligent Computing and Control Systems, ICCS 2019, Iciccs, pp. 963–968.

[4]. Kathrine, G.J.W., Praise, P.M., Rose, A.A., Kalaivani, E.C., 2019. "Variants of phishing attacks and their detection techniques", Proceedings of the international Conference on Trends in Electronics and Informatics, ICOEI 2019, Icoei, pp. 255–259.

[5]. Benavides, E., Fuertes, W., Sanchez, S., Sanchez, M., 2020. "Classification of phishing attack solutions by employing deep learning techniques: a systematic literature review". In: Rocha, Á., Pereira, R. (eds) Developments and Advances in Defense and Security. Smart Innovation, Systems and Technologies, vol 152. Springer, Singapore.

[6]. Arshad, A, Rehman, A.U., Javaid, S., Ali, T.M., Sheikh, J.A., Azeem, M., 2021. "A Systematic Literature Review on Phishing and Anti-Phishing Techniques.".

[7]. Shantanu, B. Janet and R. Joshua Arul Kumar, "Malicious URL Detection: A Comparative Study," 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), Coimbatore, India, 2021, pp. 1147-1151, doi: 10.1109/ICAIS50930.2021.9396014.

[8]. https://www.kaggle.com/code/anseldsouza/phishing-url-classification-using-knn-and-lr/input

[9]. Rakesh Verma, "What's in a URL: Fast Feature Extraction and Malicious URL Detection", ACM ISBN 978-1-4503-4909-3/17/03

[10]. Ali, Jehad, Rehanullah & Ahmad, Nasir & Maqsood, Imran. (2012). "Random Forests and Decision Trees", International Journal of Computer Science Issues (IJCSI).

[11]. Chen, Tianqi & Guestrin, Carlos (2016)."XGBoost: A Scalable Tree Boosting System". pp. 785-794. 10.1145/2939672.2939785.

[12]. Parmar, Sharad, 2020 "Detection of Phishing URL using Ensemble Learning Techniques" Master's thesis, Dublin, National College of Ireland.