ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 12, Issue 2, February 2025

Comparative Analysis of Machine Learning Models for Phishing Detection in URLs, Emails, and Webpage Content

[1] Niveaditha VR, [2] Akhil Sachin, [3] Dr. S Baghavathi Priya

[1] [2] [3] Department of Computer Science and Engineering, Amrita School of Computing, Amrita Vishwa Vidyapeetham, Chennai, India

Corresponding Author Email: [1] vrnivi03@gmail.com, [2] akhilsachin0806@gmail.com, [3] s baghavathipriya@ch.amrita.edu

Abstract*— This paper investigates and compares the performances of three machine-learning models: Random Forest, XGBoost, and the Multi-Layer Perceptron, which is often referred to as MLP. As regards evaluation criteria, the mean performances were assessed in terms of f1-score, precision, and recall. The results show that, although Random Forest and XGBoost achieved almost perfect precision and recall, the MLP achieved a higher overall f1-score, indicating a superior balance between precision and recall. This comparison brings forth the tradeoffs of model selection in classification; that is, MLP is best for balanced performance, whereas Random Forest and XGBoost excel if false positives and false negatives are to be minimized.

Index Terms*—Machine Learning, Classification, Random Forest, XGBoost, Multi-Layer Perceptron, F1-Score, Precision, Recall, Model Performance Comparison.

I. INTRODUCTION

The rapid advancement of the internet and digital communications has fundamentally transformed the way people interact, conduct business, and manage their daily activities. Although there are many advantages to these innovations, they have also made cybersecurity more vulnerable to threats, phishing attempts being one of the most prevalent and destructive. Phishing is a dishonest tactic whereby perpetrators assume the identity of reliable organizations in an attempt to coerce victims into disclosing private information, including personal identification numbers, login credentials, or financial information. These assaults are increasingly complex and difficult for standard security measures to identify and prevent. They frequently take the shape of infected files, deceptive websites, or bogus emails.

As phishing techniques evolve, attackers are employing more advanced strategies to bypass standard security protocols. For example, phishing websites may closely mimic legitimate ones, with minor variations in the URL or slight modifications in the website design that can easily go unnoticed by an unsuspecting user. Similarly, phishing emails may appear to come from credible sources, with convincing content and professional formatting that makes them difficult to distinguish from genuine communications. Additionally, malicious PDFs or other downloadable files can be embedded with harmful scripts or hidden content that, when opened, can compromise a user's device or network.

Given the growing complexity and frequency of these attacks, there is an urgent need for more sophisticated detection mechanisms that can identify and neutralize phishing threats before they cause harm. Traditional security solutions, such as basic email filters or blacklists of known malicious URLs, are no longer sufficient. These methods often rely on static rules or outdated databases, making them ineffective against newly emerging phishing tactics that are designed to evade detection.

Our project seeks to address this challenge by developing a comprehensive browser extension and mobile application that leverages advanced machine learning algorithms to detect phishing websites, malicious content, phishing emails, and harmful PDF files. By incorporating multiple layers of protection, our solution aims to provide robust, real-time detection of phishing attempts, ensuring that users are alerted to potential threats before they can fall victim to them.

II. RELATED WORK

Attacks by phishing have gradually become one of the significant problems in cybersecurity, where attackers continue to make perfect methods and techniques to deliver these attacks, aiming at deceiving people into divulging sensitive information. As a result, phishing attacks continue to become sophisticated, and more conventional methods of detection based on traditional system setups cannot compete. In light of this, the use of machine learning (ML) techniques for phishing detection becomes more applied to open new avenues for enhancing detection accuracy and adaptation to emergent threats. This area of research has been enlightened by new and growing development in ML in phishing domain and URL detection. Alnemeri and Alshammari (2023) proposed an ML-based model that analyzes the characteristics of phishing websites based on the correct features to enhance the detection performance [1]. Similarly,

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 12, Issue 2, February 2025

Qasim and Flayh (2023) did a literature survey on the various types of phishing detection techniques and discussed the advantages and disadvantages of numerous ML approaches with a focus on focusing adaptable models representing the dynamic nature of phishing attacks [2]. In addition to that, Choudhary et al. in 2023 proved ensemble methods in phishing detection using multiple classification algorithms, thereby providing better results than traditional models. [3] Further in the landscape, Sattari and Montazer investigated a systematic review on intelligent phishing detection methods, which was of great value to understanding some common challenges of this field, such as feature extraction and handling imbalanced datasets [4].

Recent studies in deep learning have also been used on phishing attack detection, and some of the contributions include the advancement in the ability to detect malicious URLs. Aldakheel et al. (2023) introduced a deep learning system that detects anti-phishing URLs by incorporating URLs with deep neural networks to combat the newest threats from phishing attacks [11]. In their phishing detection model, Elsadig et al. (2022) used BERT feature extraction thus creating a more efficient picture in the identification of phishing URLs with deeper contextual data understanding [12]. Prabakaran et al. (2023) added to the research in deep learning as they implemented variational autoencoders for the detection of phishing URLs and emphasized the possibility of using deep learning mechanisms to make the detection more precise [16]. Thus, these comparative analyses of machine learning techniques have thus been really precious in giving views of phishing detection. For example, Nagy et al. (2023) did a comparative analysis about the application of sequential and parallel ML techniques in detecting phishing URLs and concluded that sequential models usually outperform parallel methods with regard to accuracy and efficiency [10].

Pawar and Tijare (2023) further contributed to this discussion by reviewing various machine learning approaches, pointing out the key advancements in the field while also identifying areas where further research is needed [5]. Moreover, Sattari and Montazer's (2023) review of intelligent phishing website detection methods emphasized the need for adaptive systems that can evolve in tandem with increasingly sophisticated phishing techniques [4]. Zhou et al. (2023) introduced a novel phishing website detection model based on LightGBM and domain name features, demonstrating notable improvements in phishing detection by focusing on domain-level analysis [14]. Emerging technologies such as federated learning and transfer learning have also been explored in phishing detection. Thapa et al. (2023) evaluated the role of federated learning in phishing email detection, showing its potential to enhance privacy and security while maintaining strong detection performance [17].

Pawar and Tijara (2023) also contributed by summarizing different machine learning plans, highlighting the major progresses in the field plus citing some areas that require more investigation [5]. With more sophisticated methods of phishing, the Sattari and Montazer (2023) study of smart phishing website detection, instead presented the need for intelligent adaptive systems that can adapt with the rapid change in phishing technique [4]. In addition, Zhou and et al. (2023) proposed an orthogonal phishing website detection model using LightGBM and domain name features, a research which reached out to domain-level analysis and gave clear proofs of progress in phishing detection [14]. Phishing detection technologies based on federated learning and transfer learning have been the subject of many studies. Thapa et al, (2023) assessed how the technology of federated learning can be utilized in phishing email detection, which hinted at the privacy factor as it appears in action with security and strong detection performance gains [17].

Applying transfer learning in phishing detection as part of an AI-based healthcare cybersecurity system, highlights the flexibility of transfer learning in adapting to various contexts beyond traditional phishing detection [13]. The practical challenges of implementing phishing detection in real-world environments have been researched by detecting phishing URL using login URLs, which highlighted the complexities of deploying phishing detection mechanisms at scale [9]. A comprehensive survey from 2023 by Asiri et al on the intelligent designs used for detecting phishing HTML URL attacks, address the dynamic nature of such environments and the difficulties in developing effective systems [21]. N. Kanagavalli, S. Baghavathi Priya have presented HDL-FND technique which gained highly effective identification and capabilities of classification of fake information. Additionally, the HDLFND technique is comprised of a threestep procedure comprising of pre processing, feature extraction, and Bi-Directional Long Short Term Memory (BiLSTM) based classification. The research output produced improved performance of the HDL-FND technique more than the stateof-the-art techniques in different aspects [22].They have also suggested RDL-FAFND model that comprises an ensemble of the machine learning(ML) models with diverse linguistic features (EML-LF) used to classify whether the text was original or spurious. A wide range of experiments have been performed to show the prominence of the RDL-FAFND model, and it has been claimed that the proposed RDL-FAFND model is better than the state-of-the-art methods. [23]

Across these studies, it is evident that machine learning offers promising solutions for phishing detection, though significant challenges remain. Issues such as imbalanced datasets, the constantly evolving and increasingly sophisticated phishing tactics, and the scalability of detection systems in real-world applications continue to pose challenges. Research efforts focusing on developing more adaptive and scalable models to address these challenges while continuing to improve detection accuracy and reduce false positives is a security priority observed by organizations

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 12, Issue 2, February 2025

and individuals globally.

III. METHODOLOGY

The goal of this project is to build a browser extension that detects phishing websites, malicious webpage content, phishing emails, and suspicious PDF files using machine learning models. By training models to classify URLs, webpage content, and email data, this system offers real-time protection for users as they browse websites, access emails, or download files. The chosen machine learning models—Random Forest, XGBoost, and Multi-Layer Perceptron (MLP)—are used for their ability to handle different types of data efficiently and produce accurate results. The figure below shows the architecture of the proposed system.

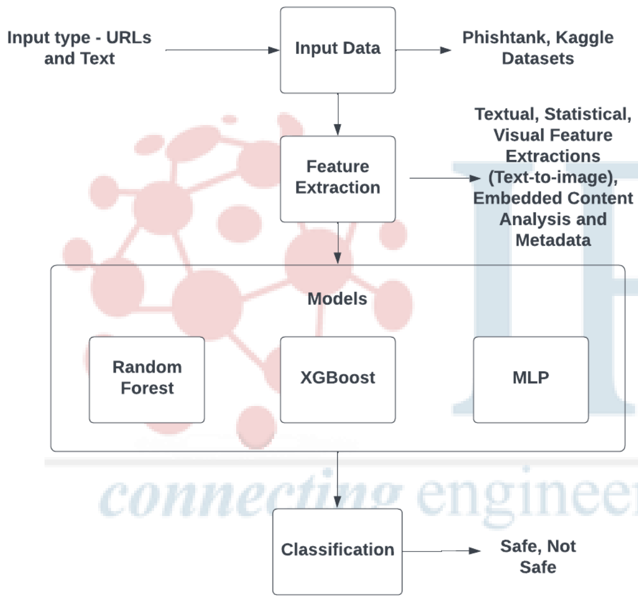
Input type - URLs and Text
Input Data
Phishtank, Kaggle Datasets
Feature Extraction
Textual, Statistical, Visual Feature Extractions (Text-to-image), Embedded Content Analysis and Metadata
Models
Random Forest
XGBoost
MLP
Classification
Safe, Not Safe

Fig. 1: System Architecture

A. Data Collection

a. Phishing and Legitimate URLs: A large dataset of phishing and legitimate URLs will be collected from trusted sources like PhishTank, OpenPhish, and Alexa Top Sites.
b. Webpage Content: Scraped content from phishing and legitimate websites will be used to train content detection models.
c. Email Content: A dataset of phishing and legitimate email bodies will be collected to classify phishing emails.
d. PDF Files: Suspicious and legitimate PDFs will be collected from email attachments, online repositories, or generated synthetically.

B. Data Preprocessing

a. URL Cleaning: URLs will be cleaned by extracting the domain and removing parameters, tracking codes, and fragments that don't contribute to phishing detection.
b. Domain extraction: Only the domain name (e.g., youtube.com) will be used for phishing classification. Normalization: URL encoding and unnecessary query parameters will be stripped.
c. Content Cleaning: Webpage and email content will be cleaned by removing HTML tags, scripts, and noninformative characters.
d. Text Normalization: Lowercasing, removing stopwords, and stemming/lemmatization of webpage and email text.

C. TF-IDF Vectorization

a. URLs: Every cleaned URL will be transformed into a vector using TF-IDF (Term Frequency-Inverse Document Frequency) so that it captures the uniqueness of words within the URLs.
b. Webpage and Email Content: The web page and email text content will be vectorized using TF-IDF for converting the contents into numbers.
c. Handling Class Imbalance: Phishing data, in most cases, is imbalanced, and there are fewer phishing samples than legitimate. SMOTE Synthetic Minority Over-sampling Technique will be applied by creating synthetic samples for the minority class in order to balance the dataset.

D. Feature Engineering

Additional features used are URL length, number of special characters, number of dots, presence of HTTPS, subdomain count, and suspicious keywords. For webpages and emails, n-grams (bigrams and trigrams) and keyword frequency will be used for better content analysis. Metadata analysis of PDFs, the number of embedded links, JavaScript presence, and external references in the PDF will be considered.

E. Model Selection

Multiple machine learning models will be trained for each task (URL, content, email, PDF):

1) Random Forest Classifier:

Random Forest Classifier is an ensemble learning algorithm that utilizes multiple decision trees together to improve classification accuracy. Each tree within the model is trained on a random part of the training data, after which, the final prediction is made based on the majority vote or the average from all decision trees. The parameters of Random Forest that were considered and used in our project are:

- n-estimators: The amount of trees in the forest. Incrementing this value improves the model's accuracy but increases computation time.
- max-depth: The maximum depth of each decision tree. Keeping a limit on the depth prevents overfitting by controlling the complexity of the trees.
- min-samples-split: This is the minimum of samples needed for an internal node to be split. Higher values make the model less sensitive to noise.
- bootstrap: Whether samples are drawn with

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 12, Issue 2, February 2025

replacement during training. Using bootstrapping helps in reducing variance.

The algorithm determines which branch on a node has a higher probability of occurring by calculating the Gini of each branch using the class and probability. This process is known as Gini Impurity.

$C\,G(t) = 1-\sum^{x}_{i=1} p^2_i$

Where,

• G(t) is the Gini impurity at node t,

• C is the number of classes (in this case, 2: phishing and safe),

• p_i is the proportion of samples of class i at node t.

The final prediction in Random Forest is based on majority voting across decision trees:

$\hat{y} = \text{mode}\{T_1(x), T_2(x), ..., T_n(x)\}$

Where,

• $T_i(x)$ is the prediction from the i-th tree, • $\hat{y}$ is the final prediction (phishing or safe).

2) XGBoost:

A potent gradient-boosting method that has been enhanced for speed and efficiency is called XGBoost. It sequentially builds trees to correct the errors of previous trees, focusing more on hard-to-predict instances. XGBoost with its ability to handle large datasets was chosen due to its high efficiency in phishing detection, and its gradient boosting mechanism helps to identify complex patterns in URLs and content. The parameters of XGBoost that were considered and used in our project are:

• n-estimators: The number of trees to be trained.

• max-depth: The maximum depth of the trees. Deeper trees allow the model to capture more complex relationships in the data.

• learning-rate: The step size reduction for each boosting step. Smaller values slow down the learning process but can lead to better performance.

• gamma: The lowest amount of loss reduction needed in order to divide. The algorithm becomes more conservative with higher values.

• subsample: The fraction of samples to use for each tree. Using a fraction helps in preventing overfitting.

XGBoost optimizes the model by minimizing a loss function through gradient descent. It achieves that by optimizing the following objective function, which consists of a loss function and a regularization term:

nK

$L(\theta) = {}^{X} l(\hat{y}_i, y_i) + {}^{X}\Omega(f_k)$

$i=1\,k=1$

Where,

• $l(\hat{y}_i, y_i)$is the loss function (e.g., binary cross-entropy for classification),

• $\Omega(f_k)$is the regularization term that controls model complexity, preventing overfitting.

After optimizing in each step it updates the prediction iteratively using gradient boosting. The prediction at step t+1 is:

$$\hat{y}_i^{(t+1)} = \hat{y}_i^{(t)} + \eta f_{t+1}(x_i)$$

where, • $\hat{y}_i^{(t)}$ is the prediction at iteration t,

• η is the learning rate

• $f_{t+1}(x_i)$is the new tree added to minimize the residual errors.

At the end it includes regularization terms $\Omega(f)$ for controlling model complexity:

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2$$

Where,

• T is the number of leaves in the tree,

• w_j are the leaf weights, • γ and λ are regularization parameters.

3) MLP (Multi-Layer Perceptron):

MLP is a custom neural network (implemented using PyTorch) used to classify webpage and email content as well as pdf analysis because it can model complex textual patterns and metadata features that are often hard to capture with simpler algorithms. MLP's adaptability to different kinds of data, such as text from emails and PDFs, makes it a versatile and efficient choice. The parameters of MLP that were considered and used in our project are:

• hidden-layer-sizes: The size of hidden layers refer to the number of neurons in each hidden layer. Larger sizes allow the model to learn more complex relationships but require more training data.

• learning-rate: The step size for weight updates. Smaller learning rates lead to slower but more precise learning.

• dropout: Neurons are randomly deactivated during training to prevent overfitting as a regularization method.

• activation function: ReLU is commonly used for hidden layers to introduce non-linearity, and Sigmoid is used in the output layer for binary classification. ReLu activation function:

$\text{ReLU}(z) = \max(0, z)$

Sigmoid activation function:

$$\text{Sigmoid}(z) = \frac{1}{1+e^{-z}}$$

MLP is trained using back propagation and gradient descent. It consists of multiple layers each applying a linear transformation followed by a non-linear activation function. For a single layer the linear transformation is:

$z = Wx + b$

Where,

• z is the output of the linear transformation, • W is the weight matrix,

• b is the bias term.

And to update the weight it uses Gradient Descent to minimize the loss using the formula below:

$$W_{new} = W_{old} - \eta \frac{\partial L}{\partial W}$$

Where,

- η is the learning rate,
- $\frac{\partial L}{\partial W}$ is the gradient of the loss function with respect to the weights.

F. Training and Evaluation

1) Model Training:

- TF-IDF Vectorization: To transform text data into numerical form, URLs, webpage content, and email data are vectorized using the TF-IDF technique.
- SMOTE: To create artificial cases for the minority class, the SMOTE technique is utilized, as phishing datasets are frequently unbalanced. Grid Search and Cross-Validation: Hyperparameter tuning is carried out using Grid Search and cross-validation to find the best-performing parameters for each model.
- Model Fitting: The vectorized data is used to train the Random Forest, XGBoost, and MLP models. Crossvalidation is used to assess the models' performance.

2) Evaluation Metrics:

Metrics including accuracy, precision, recall, and F1 score will be used to assess the models and determine how well they perform in phishing detection. False positives—legitimate websites reported as phishing—and false negatives—phishing websites reported as legitimate—will receive extra consideration.

- Accuracy: The percentage of correct predictions (both phishing and non-phishing).
- Precision: The proportion of true phishing detections out of all predicted phishing detections.
- Recall: The proportion of actual phishing cases that were correctly detected.
- F1 Score: The harmonic mean of precision and recall, used as a balanced metric when there's a class imbalance.
- ROC-AUC Score: This metric is used to evaluate the trade-off between true positive and false positive rates.

G. Model Optimization

a.Threshold Tuning: The decision threshold for classification (default 0.5) will be adjusted to minimize false positives.

b. Regularization: Regularization techniques will be applied to prevent overfitting, ensuring the models generalize well to new data.

c.Whitelisting Safe Sites: Known legitimate websites like youtube.com, google.com, and facebook.com will be whitelisted to prevent false positives.

H. System Architecture

The system will be composed of the following components:

a. Frontend (Chrome Extension):

The extension will provide a user interface where users can scan URLs, webpage content, emails, and PDFs. Interaction with the Browser: The extension will extract the URL, webpage content, or email data and send it to the backend for analysis. Whitelisting: The extension will automatically classify known safe sites without sending them for scanning.

b. Backend (Flask API):

The backend, implemented in Flask, will handle requests from the extension. It will load the pre-trained models (Random Forest, XGBoost, MLP) to make real-time predictions on URLs, content, and emails. JSON Responses: The backend will return phishing status as JSON objects to the extension for display.

I. Deployment and Integration

a. Model Deployment: The trained models will be serialized using joblib and served via a Flask API.
b. Extension Deployment: The Chrome extension will be deployed by loading it through the Chrome Extension interface, where users can access it for real-time phishing detection.

J. Testing and Validation

a. Cross-Browser Testing: The extension will be tested across multiple browsers (Chrome, Firefox, etc.) to ensure compatibility.
b. End-to-End Testing: The complete system (frontend extension + backend Flask API) will be tested using both phishing and legitimate samples to ensure accuracy and user experience.
c. Edge Case Handling: Special attention will be given to handling edge cases like incomplete URLs, pages with minimal content, and PDFs with varying structures.

K. User Interface

The extension interface will be designed with usability in mind, providing users with simple buttons to trigger scans and displaying results in a clear and concise manner. Visual cues (e.g., green for safe, red for phishing) will be used to indicate the results of the scan.

L. Monitoring and Updating

The models will be periodically updated with new phishing samples to keep them relevant. User feedback on false positives or negatives will be incorporated into future updates.

IV. RESULTS

This work presents evaluation of the performance of machine learning models, including Random Forest,

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 12, Issue 2, February 2025

XGBoost, and MLP, on the three selected phishing detection, namely, URL, Email, and Content classification. A comparative analysis was performed based on precision, recall, and F1 score to determine the overall effectiveness of the models for the different phishing detection tasks. The detailed results for each of the tasks are presented below.

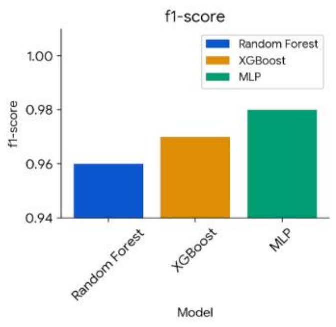

Fig. 2: URL - F1-Scores

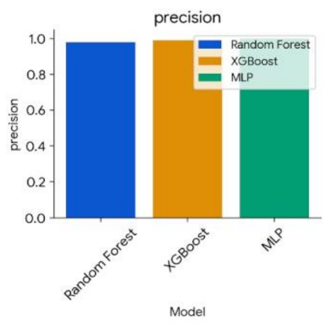

Fig. 3: URL - Precision Scores

A. URL Phishing Detection

The models give high performance for phishing URL detection in all the metrics. Random Forest and XGBoost perform almost perfectly, showing high precision and recall, while MLP has about balanced performance with a slightly low F1 score but with quite competitive performance altogether.

1) *Random Forest:* Precision: 0.98, Recall: 0.96, F1 score: 0.97, Accuracy: 0.97

2) *XGBoost:* Precision: 0.99, Recall: 0.97, F1 score: 0.98, Accuracy: 0.98

3) *MLP:* Precision: 0.98, Recall: 0.98, F1 score: 0.98, Accuracy: 0.98 In the contest, XGBoost soared ahead of the rest with a high F1 score and accuracy. MLP also is seen well-matched in worth, for in its metrics appear rather wellbalanced, thus making it a strong contender for URL phishing detection.

B. Detection of Email Phishing

All the models here performed remarkably well in the task of email phishing detection, but MLP performed with the best quality on all performance metrics. Random Forest and XGBoost showed similar results, while MLP reached the highest F1 scores with a good balance of precision and recall.

1) *Random Forest:* Precision: 0.96, Recall: 0.96, F1-score: 0.96, Accuracy: 0.96

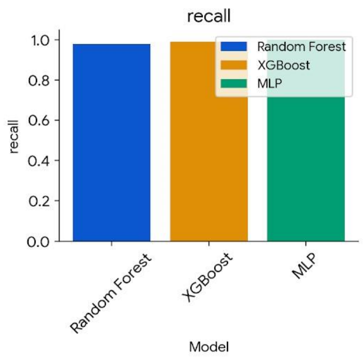

Fig. 4: URL - Recall Scores

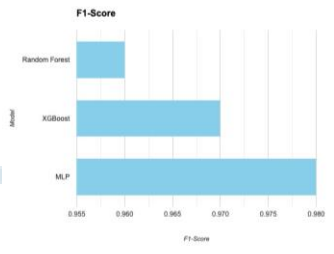

Fig. 5: Email - F1-Scores

2) *XGBoost:* Precision: 0.97, Recall: 0.97, F1-score: 0.97, Accuracy: 0.97

3) MLP: Precision: 0.98, Recall: 0.98, F1-score: 0.98, Accuracy: 0.98

Thus, for email phishing detection, which got the highest F1-score of 0.98, the MLP model showed this task's best outcomes compared to Random Forest and XGBoost. Therefore, as shown by MLP, maintaining high precision and recall is vital in this task, where a balanced performance is required.

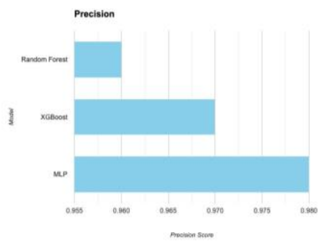

Fig. 6: Email - Precision Scores

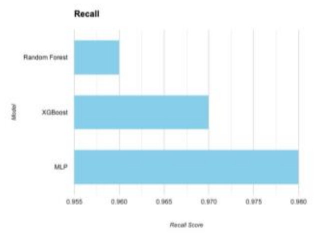

Fig. 7: Email - Recall Scores

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 12, Issue 2, February 2025

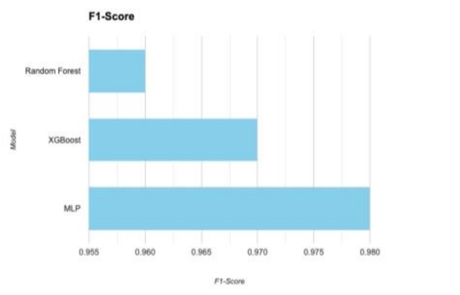

Fig. 8: Content - F1-Scores

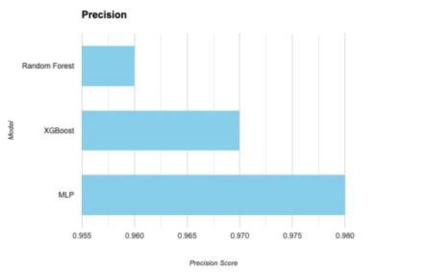

Fig. 9: Content - Precision Scores

C. Phishing detection based on webpage content

With regard to webpage content classification, the results were consistent across models, being almost similar to those in email phishing detection. In terms of precision, recall, and F1-score, MLP yet again outperformed all other models.

1) *Random Forest:* Precision: 0.96, Recall: 0.96, F1-score: 0.96, Accuracy: 0.96.

2) *XGBoost:* Precision:0.97, Recall: 0.97, F1-score: 0.97, Accuracy: 0.97.

3) *MLP:* Precision:0.98, Recall: 0.98, F1-Score: 0.98, Accuracy: 0.98.

The best performance again went to MLP with an F1-score of 0.98, followed closely by XGBoost and Random Forest. The elegant balance of precision and recall puts the MLP slightly ahead into another dimension; it seems to be much better equipped to cope with difficult tasks like detecting phishing contents.

V. CONCLUSION AND FUTURE WORK

Some of the grounds on which future research and improvements could be carried out are as follows:

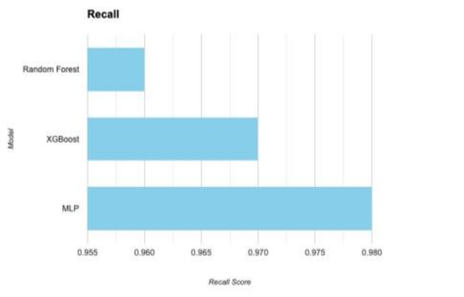

Fig. 10: Content Recall Scores

A. Generalization and Transferability of the Model

The capability of the system to detect phishing over larger varieties of languages and regions must be improved. Phishing websites in languages other than English present a unique challenge wherein future work may need to deal with using multilingual datasets and employing techniques like transfer learning to allow models to perform well across languages and cultural contexts.

B. Federated Learning for Privacy-Preserving Detection

Future versions for enhanced privacy of the user could also make use of federated learning wherein the model is trained in multiple decentralized devices preserving the sensitive user data of users. This will enable the model to become more and more robust without compromising the anonymity of the users.

REFERENCES

[1] Shouq Mohsen Alnemeri, Majid Alshammari, Applied Sciences (2023). Detecting Phishing Domains Using Machine Learning,

[2] Marwa Abd Al Hussein Qasim, Dr. Nahla Abbas Flayh, Wasit Journal for Pure Sciences (2023), Phishing Website Detection Using Machine Learning: A Review.

[3] Tarun Choudhary, Siddhesh Mhapankar, Rohit Bhddha, Ashish Kharuk, and Dr. Rohini Patil, Journal of artificial intelligence and technology (2023). Machine Learning Approach for Phishing Attack Detection.

[4] Yeganeh Sattari, GholamAli Montazer, Research Square, 31 Jan (2023) Intelligent Methods in Phishing Website Detection: A Systematic Literature Review.

[5] Nikita Pawar, Dr. P. A. Tijare, International journal of scientific research in computer science, engineering and information technology, 2023. A Review on Phishing Website Detection Using Machine Learning Approach.

[6] Chengge Duan, Minze Wang, Xin Lu, Junming Wang (2023), Academic journal of computing and information science. A phishing website detection system based on machine learning methods

[7] Yohan Muliono, Muhammad Amar Ma'ruf, Zakiyyah Mutiara Azzahra, Engineering, Mathematics and Computer Science Journal (EMACS), Phishing Site Detection Classification Model Using MachineLearning Approach

[8] Swatej Patil, Mayur Patil, KotadiChinnaiah, 4th International Conference on Machine Learning, Image Processing, Network Security and Data Sciences (MIND-2022), 2023. Machine Learning and Deep Learning for Phishing Page Detection,

[9] Manuel Sanchez-Paniagua, Eduardo Fidalgo Fern´ andez, Enrique Alegre,´ Wesam Al-Nabki, And V´ıctor Gonzalez-Castro -IEEE Access, 2022.´ Phishing URL Detection: A Real-Case Scenario Through Login URLs.

[10] Naya Nagy, Malak Aljabri, Afrah Shaahid, Amnah Albin Ahmed, Fatima Alnasser, Linda Almakramy, Manar Alhadaband Shahad Alfaddagh, Phishing URLs Detection Using Sequential and Parallel MLTechniques: Comparative Analysis

[11] Eman Abdullah Aldakheel, Mohammed Zakariah,

Ghada Abdalaziz Gashgari, Fahdah A. Almarshad and Abdullah I. A. Alzahrani, 2023. A Deep Learning-Based Innovative Technique for Phishing Detection in Modern Security with Uniform Resource Locators,

[12] Muna Elsadig, Ashraf Osman Ibrahim, Shakila Basheer, Manal Abdullah Alohali, Sara Alshunaifi, Haya Alqahtani, Nihal Alharbi and WamdaNagmeldin, 2022. Intelligent Deep Machine Learning Cyber Phishing URL Detection Based on BERT Features Extraction,

[13] Chinmay Chakraborty, Senthil Murugan Nagarajan, Ganesh Gopal Devarajan, T V Ramana, Rajanikanta Mohanty, ACM Transactions on Sensor Networks, 2023. Intelligent AI-based Healthcare Cyber Security System using Multi-Source Transfer Learning Method,

[14] Jingxian Zhou, Haibin Cui, Xina Li, Wenjin Yang and Xi Wu, Symmetry 2023. A Novel Phishing Website Detection Model Based on LightGBM and Domain Name Features

[15] Lohith Ranganatha Reddy Kandula, T. Jaya Lakshmi1, Kalavathi Alla, Rohit Chivukula, International Journal of Safety and Security Engineering, 2022. An Intelligent Prediction of Phishing URLs Using ML Algorithms,

[16] Manoj Kumar Prabakaran, Parvathy Meenakshi Sundaram, Abinaya Devi Chandrasekar, IET Information Security, 2023. An enhanced deep learning-based phishing detection mechanism to effectively identify malicious URLs using variational autoencoders,

[17] Chandra Thapa, Jun Wen Tang 2, Alsharif Abuadbba, Yansong Gao, Seyit Camtepe, Surya Nepal, Mahathir Almashor and Yifeng Zheng, Sensors – (MDPI 2023). Evaluation of Federated Learning in Phishing Email Detection,

[18] Ala Mughaid, Shadi AlZu'bi, Adnan Hnaif, Salah Taamneh, Asma Alnajjar, Esraa Abu Elsoud, Cluster Computing(2022). An intelligent cyber security phishing detection system using deep learning techniques.

[19] Giovanni Apruzzese, Mauro Conti, Ying Yuan, Annual Computer Security Applications Conference (ACSAC)'22. SpacePhish: The Evasionspace of Adversarial Attacks against Phishing Website Detectors using Machine Learning

[20] A.G. Sreedevi, T. Rama Rao (2019). Reinforcement learning algorithm for 5G indoor device-to-device communications

[21] Sultan Asiri, Yang Xiao, Saleh Alzahrani, Shuhui Li, And Tieshan Li, IEEE Access, 2023. A Survey of Intelligent Detection Designs of HTML URL Phishing Attacks.

[22] N. Kanagavalli, S. B. Priya and J. D,6th International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2022. Design of Hyperparameter Tuned Deep Learning based Automated Fake News Detection in Social Networking Data.

[23] Kanagavalli, N, Priya, S. Baghavathi,Intelligent Automation and Soft Computing(2022) Social Networks Fake Account and Fake News Identification with Reliable Deep Learning.