International Journal of Engineering Research and Science & Technology

www.ijerst.org

ISSN : 2319-5991

Vol. 21 No. 3 (1) 2025

ijerst.editor@gmail.com

editor@ijerst.com

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com
Vol. 21, No. 3(1), 2025

Research Paper

Optimized Feature Selection and Enhanced Phishing Detection Using Multi-Objective Techniques and XGBoost

A. Mamatha[1], A. Srikanth[2], K. Shoba[2], S. Mahendra Varma[2], Amgoth Shiva[2]
[1]Assistant Professor, [2]UG Student, [1,2]Department of Computer Science and Engineering (CSIT)
[1,2]Sree Dattha Institute of Engineering and Science, Sheriguda, Ibrahimpatnam, 501510, Telangana.

ABSTRACT

Web spoofing attacks pose a serious threat to the confidentiality and integrity of online interactions, often tricking users into disclosing sensitive information by mimicking legitimate websites. While existing server-side defenses—such as SSL/TLS protocols and domain validation—offer some degree of protection, they are inherently reactive and frequently fall short in real-time threat mitigation. These approaches often fail to detect spoofed sites promptly and accurately, leading to both false positives and undetected threats. Given the evolving sophistication of spoofing techniques, a proactive client-side detection system is essential for timely and effective protection. In response to this gap, the proposed pishcatcher system introduces a machine learning-driven client-side defense mechanism designed to detect and block web spoofing attempts in real time. By extracting and analyzing features from web pages—such as HTML structure, CSS layout, and JavaScript behaviors—the system utilizes advanced classification algorithms to differentiate between genuine and malicious websites. Furthermore, its adaptive learning capability allows continuous improvement in detecting emerging spoofing patterns, ensuring resilience against evolving attack vectors. This approach not only enhances the accuracy and responsiveness of spoofing detection but also provides users with a reliable tool to safeguard online interactions before damage occurs.

Keywords: Web spoofing, phishing detection, client-side security, machine learning, HTML analysis, real-time protection, pishcatcher system.

Received: 14-7-2025 Accepted: 21-8-2025 Published: 28-8-2025

1. INTRODUCTION

Phishing attacks have become a dominant cybersecurity threat, exploiting user trust and the ubiquity of digital services to deceive individuals into revealing sensitive information. In India, where internet usage is expanding rapidly—with over 918 million subscribers and increasing rural penetration—the attack surface has grown significantly. Financial phishing alone saw a 175% increase in early 2024, highlighting a concerning trend in cybercrime targeting banking, e-commerce, government portals, and healthcare services. These attacks have evolved from simplistic static email scams into dynamic, AI-powered campaigns that employ homograph techniques, Unicode spoofing, and realistic cloned websites to bypass traditional detection methods such as blacklist filtering or heuristic-based systems. Current solutions often suffer from latency, false positives, or inadequate adaptability to novel attack vectors, leaving end users and critical infrastructure vulnerable to significant financial and reputational damage.To address this gap, our research introduces pishcatcher, an advanced, client-side phishing detection framework designed to operate in real-time within a user's browser environment. It uses a hybrid machine learning pipeline that combines a Multilayer Perceptron (MLP) for deep feature extraction from raw URL tokens and character-level patterns, with a Random Forest classifier that enhances decision accuracy and interpretability. This dual-model architecture allows pishcatcher to learn from both high-level semantic features and low-level statistical cues, ensuring it can generalize across evolving spoofing strategies. Trained on an expansive, India-specific dataset that

includes English and regional-language URLs, pishcatcher is tailored to detect culturally and linguistically localized phishing attacks often missed by global models.Unlike traditional URL defenses that depend on server-side lookups or static rules, pishcatcher functions entirely on-device, delivering sub-50ms detection latency through optimized asynchronous processing and local inference caching. This significantly reduces the attack window and provides users with instant, actionable feedback—flagging suspicious domains, blocking access to spoofed pages, and issuing clear confidence-based warnings. It also adapts to emerging threats through continuous learning, making it resilient against AI-generated phishing schemes, browser-in-the-browser deception, and obfuscation methods used to bypass static filters.The system is especially valuable for applications in sectors with high sensitivity and frequent attack exposure. In banking and online financial services, it helps protect users from fraudulent portals that mimic login pages or transaction gateways. On e-commerce platforms**,** pishcatcher blocks fake product pages and malicious payment URLs designed to harvest credentials. In social media and email environments**,** it detects deceptive content disguised as legitimate login prompts or shared links. It also safeguards government and healthcare platforms**,** where breaches can lead to large-scale data leaks and operational paralysis, as seen in the AIIMS New Delhi cyberattack.

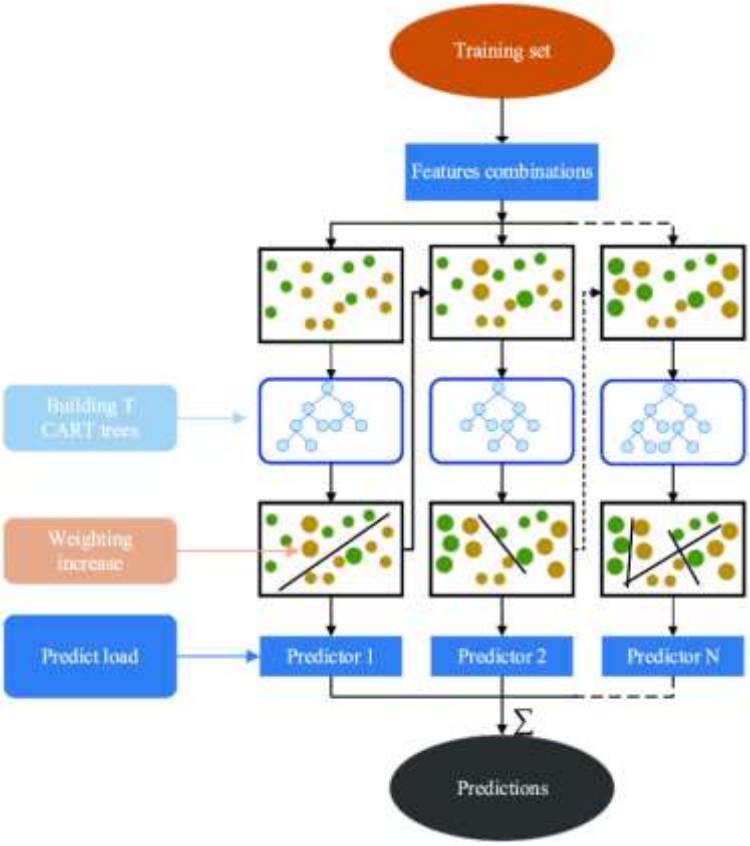

Fig. 1: Schematic illustration of the XGBOOST model.

From a broader perspective, the project addresses a critical intersection of technological innovation, user safety, and public trust in India's growing digital economy—projected to exceed $1 trillion by 2025. As regulatory frameworks such as the Personal Data Protection Bill demand more transparency and accountability, pishcatcher's explainable AI model becomes even more significant. By presenting clear, interpretable alerts backed by confidence scores and visual cues, it fosters user trust and compliance with national cybersecurity mandates.In summary, this research not only presents a technically robust, low-latency, and scalable phishing detection solution but also fills critical gaps in current defenses by considering the socio-linguistic nuances of Indian internet users. It represents a major step forward in building proactive, intelligent security tools that adapt with the threat landscape

while maintaining seamless user experience. pishcatcher ultimately empowers individuals, enterprises, and public institutions to navigate the digital world safely, laying the groundwork for a more secure and resilient cyber ecosystem.

2. LITERATURE SURVEY

Proposed an anti-phishing architecture using machine learning approaches. The selected features were categorized into URL obfuscation features, third-party features, and hyperlink-based features. They used a dataset of 2119 phishing sites from Phish Tank and 1407 legitimate sites from the Alexa Database [1]. The algorithms used were RF, J48, LR, BN, MLP, SMO, AdaBoostM1, and SVM to calculate the metrics of sensitivity, specificity, precision, accuracy, and ERR. According to the results, RF showed a maximum accuracy of up to 0.9931 among all the other classifiers. The dataset was limited to textual objects with no captcha objects, while the attackers can use embedded objects instead of textual objects to bypass anti-phishing attacks. Suryan et al. [2] proposed a machine learning model that involved almost thirty features based on URL for the classification between phishing and legitimate websites. The dataset of the URLs was obtained from the machine learning repository provided by UCI. The dataset was then dimensionally reduced. They applied the machine learning algorithms of random forest, support vector machine, generalized linear model, generalized additive model, recursive partitioning, and regression trees on the reduced dataset. The RF algorithm with 300 trees provided an accuracy of 96.65 percent and a precision of up to 97.4 percent, being the best algorithm for the scenario. The results can be improved by applying higher-order dimensionality reduction techniques like the variance inflation factor (VIF). Al-Sarem et al. [3] presented an optimized stacking ensemble method for detecting phishing attacks based on URL features. Different machine learning classifiers, AdaBoost, XGBoost, bagging, GradientBoost, and LightGBM were used, and the generic algorithm was used as the optimizer algorithm in order to apply optimization to the mentioned ensemble machine learning methods. Three datasets of phishing data were obtained from Mendeley and UCI library for machine learning.

The optimized ensemble method improved the results, and the accuracy was found to be up to 98.58%. The datasets used for the research were limited, and the results can be more efficient using more appropriate datasets. Butnaru et al. [4] presented an SVM model of machine learning to detect and block phishing attacks with the help of only a novel combination of URL features. Naïve Bayes, decision tree, random forest, support vector machine, and multilayer perceptron were the algorithms used on the imbalanced dataset obtained from Kaggle and PhishTank. The optimized RF algorithm provided the best results with an accuracy of 99.29 percent and detected phishing attacks well as compared to the Google safe browsing (GSB). The features can be optimized and formulated as novel to enhance the effectiveness of the results. Cuzzocrea et al. [5] also came up with a website phishing detection technique using a machine learning model. They used the decision tree (DT) algorithm to check the legitimacy of the website. The dataset was obtained from the UCI library and different features of the website were selected, including URL as well as others like IP, Web Traffic, Domain Age, etc. Decision stump, logistic model tree (LMT), RepTree, random forest, Hoeffding tree and J48 algorithms were applied to the dataset. The results showed that J48 and RepTree performed better than the other algorithms. The proposed model was aimed at classifying the website as phishing or legitimate and the model performed very well. Still, the research can be pursued to make the results more effective by further focusing on feature extraction and keeping the infected code in view. Cristian de Souza et al. [6] proposed a tool that they named “PhishKiller,” proficient in identifying and mitigating phishing attacks through an approach of the proxy, engaged to catch user-accessed addresses and featureless machine learning techniques for classifying the URLs as legitimate or phisher.

The URL datasets were obtained from UCI and PhishTank and the data was cleaned. The LSTM algorithm of machine learning was used for the testing and training accompanied by the neural

networks in order to avoid further predictions as any infected URL was being added to the maintained database. The results showed up to 98.30% accuracy in the detection of legitimate or phishing websites, and the process took only 81.86 ms to identify and mitigate the malicious website. The featureless approach requires a bulk amount of data for efficient results; hence, there needs to be a well-maintained database for improving the efficiency, and this is a limitation of this featureless approach. Tharani et al. [7] conducted a study to know how the original URLs are mimicked by the phishers, and for this, they used information gain and chi-squared methods of machine learning for feature selection on a phishing dataset. The aim of their research was to identify the features of URLs that are used by phishers to make them look like the actual URLs. They obtained the phishing dataset from the Mendeley repository and extracted 48 features. The total legitimate URLs were 5000, obtained from Alexa and Common Crawl. In the same way, the same amount of phishing URLs were obtained from PhishTank and OpenPhish. The KNN and SVC algorithms were used for classifying the feature detection of URLs that cause phishing attacks. KNN performed well for up to 10 features, while KNN and SVC performed best when the features were more than 10. The results show that null self-redirect hyperlinks in URL and domain name mismatch are the most common techniques that are used to trick humans into phishing attacks. This research is only limited to identifying the most common techniques that are used by phishing attackers.Sameen et al. [8] came up with a PhishHaven model that is an AI-generated phishing URL detection system working on the basis of ensemble machine learning as well as lexical feature analysis. HTML URL encoding as lexical features, as well as the URL hit approach for the detection of tiny URLs, was also presented. The ensemble machine learning model resulted in real-time detection of phishing attacks. The final evaluation was performed by the unbiased voting method. The dataset of phishing URLs was obtained through the PhishDeep and PhishTank, while the normal URLs were obtained from Alexa. AdaBoost, bagging, decision tree, extra tree, GradientBoost, KNN, LR, NN, RF, and SVM classifiers were applied, and SVM performed best in terms of 97.68% precision. The overall PhishHaven model was up to 98% accurate. The limitation of this model is that it can only detect URLs based on lexical features and patterns similar to the DeepPhish. Alsariera et al [9].

Stated that cybercriminals seek to obtain access to a victim's personal information, such as passwords, credit card details, and other similar data by using fake websites, emails, or any other authentic online service. Although there are a lot of methods to recognize phishing websites, such as methods based on third parties, methods based on source code, and methods based on URLs, individuals still fall for fraud. According to the findings of this study, it is possible to identify phishing websites by using word embeddings that make use of plain text and domain-specific terminology that is obtained from the source code. To evaluate the model's word embeddings, ensemble and multimodal methodologies were used. In trials, they found that using multimodal analysis with domain-specific text achieved a significant TPR of 99.59%, FPR of 0.93%, and MCC of 98.68%, which contributed to an overall accuracy of 99.34%. Almalaq et al [10].

3. PROPOSED METHODOLOGY

The implementation process for phishing URL classification using the Phish_Tank_Storm dataset begins with loading the dataset containing labeled phishing and legitimate URLs, followed by handling missing values and breaking down URLs into components like domains, paths, and queries. Feature extraction involves analyzing characters and symbols within URLs, and MinMaxScaler is used to normalize the data. Exploratory Data Analysis (EDA) with bar plots, histograms, and correlation matrices helps understand feature distributions and relationships. A Multi-Layer Perceptron (MLP) Classifier is initially trained to serve as a baseline, and its performance is evaluated using metrics such as accuracy, precision, recall, F1-score, and a confusion matrix. Subsequently, a Random Forest Classifier is trained on the same data to compare its effectiveness. A novel hybrid

model is proposed by using the MLP as a deep feature extractor and feeding these representations into a Random Forest

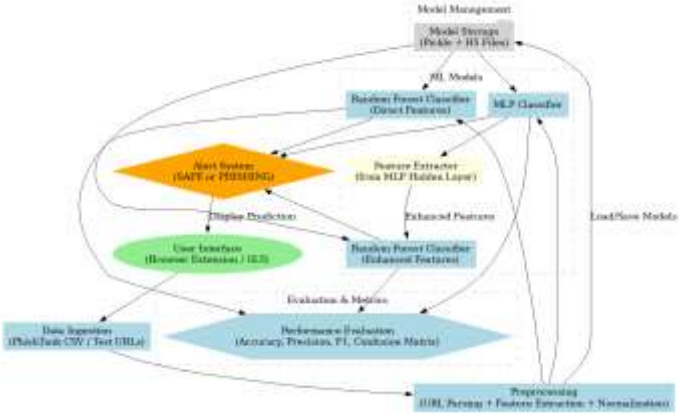

Fig. 2: Block Diagram of Proposed System.

Classifier, combining deep learning's ability to capture complex patterns with the ensemble model's classification strength. The models are compared graphically and tabularly to highlight the performance improvements of the hybrid model. Finally, real-time predictions are conducted using the trained hybrid model on new, preprocessed URLs, classifying them as legitimate or phishing, showcasing the practical application of this approach in real-world cyber threat detection.

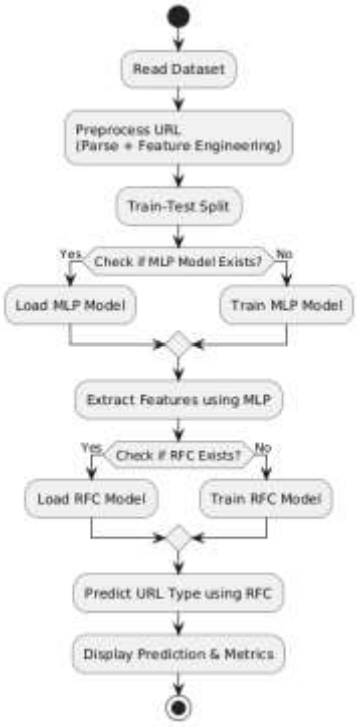

Fig. 3: Flow Diagram of MLP with RFC Classifier.

The RFC with MLP Classifier is a hybrid machine learning approach that integrates the strengths of a Multi-Layer Perceptron (MLP) and a Random Forest Classifier (RFC) to enhance phishing URL detection. In this architecture, the MLP is initially trained on numerical URL features, and instead of using its final output for classification, activations from its intermediate hidden layers are extracted to serve as deep-learned feature vectors. These high-level abstract representations are then normalized and passed to a Random Forest Classifier, which utilizes its ensemble of decision trees to perform the final classification based on majority voting. This hybrid method combines the deep learning capability of MLP for capturing complex, non-linear relationships in data with the robustness, interpretability, and resistance to overfitting offered by RFC. The approach improves overall accuracy,

handles noise more effectively, and remains scalable for large datasets, making it particularly well-suited for real-time phishing detection in diverse and dynamic cyber environments.

4. RESULTS

The implementation of the phishing detection system begins with loading the Phish_Tank_Storm dataset, which includes labeled URLs and associated numerical features essential for classification. The data undergoes preprocessing, where missing values are replaced with zeros and features are normalized using Min Max Scaler to ensure consistent scale. Exploratory Data Analysis (EDA) is then conducted to visualize class distribution and understand feature behavior through plots like bar graphs, histograms, and box plots. An initial MLP Classifier is trained on 80% of the data using ReLU and softmax activations, followed by performance evaluation using accuracy, precision, recall, F1-score, and confusion matrix. Parallelly, a Random Forest Classifier (RFC) is trained and evaluated to compare with the MLP. To improve classification accuracy, a hybrid model—RFC with MLP—is proposed, where the MLP acts as a feature extractor, and the deep-learned features are classified by the RFC using ensemble decision trees. All three models are compared through graphical analysis, highlighting the superior performance of the hybrid approach. Finally, the trained hybrid model is used to classify new URLs in real time, determining whether each URL is SAFE or PHISHING, thereby demonstrating its practical utility in web security applications.

	url	label
0	nobell.it/70ffb52d079109dca5664cce6f317373782/...	1
1	www.dghjdgf.com/paypal.co.uk/cycgi-bin/webscrc...	1
2	serviciosbys.com/paypal.cgi.bin.get-into.herf....	1
3	mail.printakid.com/www.online.americanexpress....	1
4	thewhiskeydregs.com/wp-content/themes/widescre...	1
...	...	...
96002	xbox360.ign.com/objects/850/850402.html	0
96003	games.teamxbox.com/xbox-360/1860/Dead-Space/	0
96004	www.gamespot.com/xbox360/action/deadspace/	0
96005	en.wikipedia.org/wiki/Dead_Space_(video_game)	0
96006	www.angelfire.com/goth/devilmaycrytonite/	0

96007 rows × 2 columns

Fig. 4: Upload of Phish Tank Storm Dataset and Its Analysis.

The figure 4 represents the loading and initial analysis of the Phish_Tank_Storm dataset. The dataset contains multiple URL-based features along with their classification labels. The figure shows the structured data after being loaded into the system, with columns such as ranking, mld_res, jaccard similarity scores, and label. It also includes an analysis of the data distribution, highlighting the proportion of legitimate and phishing URLs present in the dataset. This step is crucial in understanding the nature of the dataset before proceeding with preprocessing and model training.

Extracted numeric fetaures from dataset URLS

	url_length	qty_dot_url	qty_hyphen_url	qty_slash_url	qty_questionmark_url	qty_equal_url	qty_at_url	qty_and_url	qty_exclamation_url	qty_space_url
0	225	6	4	10	1	4	0	3	0	0
1	81	5	2	4	0	2	0	1	0	0
2	177	7	1	11	0	0	0	0	0	0
3	60	6	0	2	0	0	0	0	0	0
4	116	1	1	10	1	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...
96002	39	3	0	3	0	0	0	0	0	0
96003	44	2	2	4	0	0	0	0	0	0
96004	42	2	0	4	0	0	0	0	0	0
96005	45	2	0	2	0	0	0	0	0	0
96006	41	2	0	3	0	0	0	0	0	0

96007 rows × 85 columns

Fig. 5: Data Preprocessing.

The figure 5 illustrates the preprocessing steps applied to the dataset. It includes handling null values by replacing them with zeros, ensuring data consistency. Min Max Scaler is applied to normalize numerical values between 0 and 1, preventing any feature from dominating others due to scale differences. The figure also presents an overview of the transformed dataset after preprocessing, ensuring it is ready for exploratory data analysis and model training.

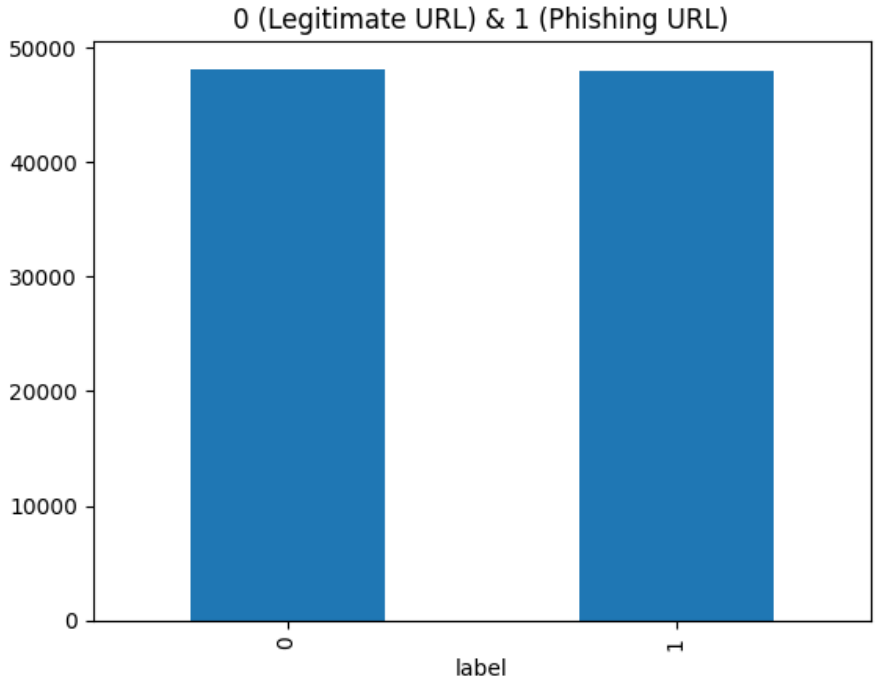

Fig. 6: Exploratory Data Analysis (EDA) Plots of the Project.

The figure 6 displays various graphical visualizations used to analyze the dataset. The number of phishing and legitimate URLs is represented in a bar plot, showing the class distribution. Additional plots, such as histograms, box plots, and correlation heatmaps, provide insights into feature distributions and relationships. The analysis helps in identifying patterns, detecting outliers, and understanding the influence of different features on classification.

International Journal of Engineering Research and Science & Technology

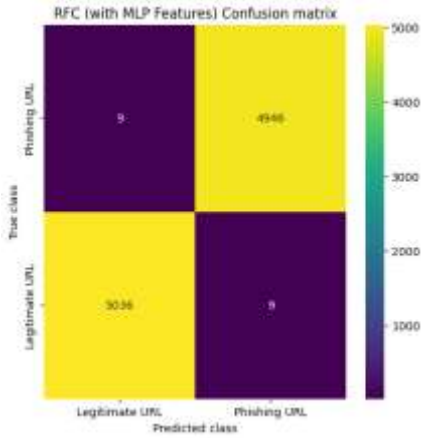

Fig. 7: Performance Metrics and Classification Scatter Plot for RFC with MLP Classifier Model.

The figure 7 demonstrates the results of the proposed hybrid model (RFC with MLP Classifier), which outperforms both previous models. The classifier achieves an exceptional accuracy of 99.82%, with precision, recall, and F1-score all reaching 99.81%. The scatter plot clearly shows a distinct separation between phishing and legitimate URLs, indicating that the model effectively utilizes deep learning-based features with Random Forest's decision-making capability to maximize classification accuracy.

Fig. 8: Model Prediction on Test Case inputs**.**

The figure 8 showcases how the trained RFC with MLP Classifier model performs on new test data. URLs from an external test dataset are processed, feature-extracted, and classified as either SAFE (legitimate) or PHISHING. The predictions are displayed along with the corresponding input URLs, demonstrating the model's real-time phishing detection capability. This step ensures that the model generalizes well to unseen data.

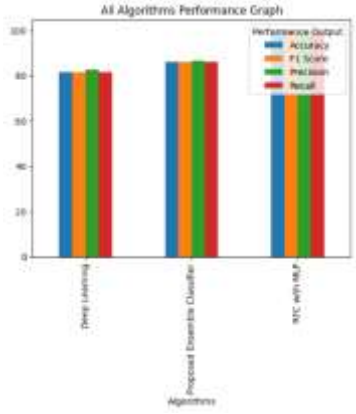

Fig. 9: Performance Comparison Graph of All Models.

The figure 9 provides a side-by-side comparison of the three classifiers: MLP Classifier, Random Forest Classifier, and RFC with MLP Classifier. The comparison is based on key performance metrics such as accuracy, precision, recall, and F1-score. The bar graph visually represents the significant improvement achieved by the proposed hybrid model, confirming its effectiveness in phishing URL classification. The RFC with MLP Classifier outperforms both existing models, making it the most suitable approach for accurate phishing detection.

5. CONCLUSION

The research successfully developed a robust machine learning-based phishing detection system using models like MLP, Random Forest, and a proposed hybrid approach combining MLP with RFC, which achieved superior performance in classifying URLs as legitimate or phishing. By extracting and analyzing URL features, the system demonstrated high accuracy, precision, recall, and F1-scores, emphasizing the effectiveness of machine learning in enhancing cybersecurity against phishing threats. Looking ahead, the future scope includes implementing adversarial training to bolster defense against obfuscated attacks, adopting continuous learning for real-time adaptation to new threats, and enabling cross-platform integration through APIs for email clients, browsers, and security tools. Further enhancements include utilizing collaborative filtering for shared intelligence, improving explainability through interpretable outputs, and embedding educational modules to foster user awareness and promote safer online behavior—ensuring the system remains scalable, transparent, and resilient in evolving cyber landscapes.

REFERENCES

[1]. Rao, R.S.; Pais, A.R. Detection of phishing websites using an efficient feature-based machine learning framework. Neural Comput. Appl. 2019, 31, 3851–3873.

[2]. Suryan, A.; Kumar, C.; Mehta, M.; Juneja, R.; Sinha, A. Learning Model for Phishing Website Detection. EAI Endorsed Trans. Scalable Inf. Syst. 2020, 7, e6.

[3]. Al-Sarem, M.; Saeed, F.; Al-Mekhlafi, Z.G.; Mohammed, B.A.; Al-Hadhrami, T.; Alshammari, M.T.; Alreshidi, A.; Alshammari, T.S. An optimized stacking ensemble model for phishing websites detection. Electronics 2021, 10, 1285.

[4]. Butnaru, A.; Mylonas, A.; Pitropakis, N. Towards lightweight url-based phishing detection. Future Internet 2021, 13, 154.

[5]. Cuzzocrea, A.; Martinelli, F.; Mercaldo, F. A machine-learning framework for supporting intelligent web-phishing detection and analysis. In Proceedings of the Proceedings of the 23rd International Database Applications & Engineering Symposium, Athens, Greece, 10–12 June 2019; pp. 1–3.

[6]. de Souza, C.H.; Lemos, M.O.O.; Dantas Silva, F.S.; Souza Alves, R.L. On detecting and mitigating phishing attacks through featureless machine learning techniques. Internet Technol. Lett. 2020, 3, e135.

[7]. Tharani, J.S.; Arachchilage, N.A.G. Understanding phishers' strategies of mimicking uniform resource locators to leverage phishing attacks: A machine learning approach. Secur. Priv. 2020, 3, e120.

[8]. Sameen, M.; Han, K.; Hwang, S.O. PhishHaven—An efficient real-time ai phishing URLs detection system. IEEE Access 2020, 8, 83425–83443.

[9]. Alsariera, Y.A.; Adeyemo, V.E.; Balogun, A.O.; Alazzawi, A.K. Ai meta-learners and extra-trees algorithm for the detection of phishing websites. IEEE Access 2020, 8, 142532–142542.

[10]. Naaz, S. Detection of Phishing in Internet of Things Using Machine Learning Approach. Int. J. Digit. Crime Forensics (IJDCF) 2021, 13, 1–15.