Advancing Phishing Detection with Random Forest: A Machine Learning Approach for Enhanced Cybersecurity

Sucheta Chandra
Dept. of Computer Application and Science
Institute of Engineering & Management,
University of Engineering & Management, Kolkata, India
Email: sucheta.chandra@iem.edu.in

Priyanka Das
Dept. of Computer Application and Science
Institute of Engineering & Management,
University of Engineering & Management, Kolkata, India
Email: priyankadas700@gmail.com

Ankita Mandal
Dept. of Computer Application and Science
Institute of Engineering & Management,
University of Engineering & Management, Kolkata, India
Email: mandalankita92@gmail.com

Sriparno Chakroborty
Dept. of Computer Application and Science
Institute of Engineering & Management,
University of Engineering & Management, Kolkata, India
Email: sriparnochakraborty.iembca2026@gmail.com

Ishika Chowdhury
Dept. of Computer Application and Science
Institute of Engineering & Management,
University of Engineering & Management, Kolkata, India
Email: ishikachowdhury.iembca2026@gmail.com

Kathamrita Ghosh
Dept. of Computer Application and Science
Institute of Engineering & Management,
University of Engineering & Management, Kolkata, India
Email: ghoshkathamrita2005@gmail.com

***Abstract*—Phishing attacks remain a critical threat to cybersecurity, leveraging deception to steal sensitive information from individuals and organizations. Traditional detection methods often fail to keep up with evolving phishing tactics, highlighting the need for advanced solutions. This paper investigates the use of the Random Forest algorithm, a powerful machine learning technique, to enhance phishing detection systems. By leveraging the ensemble learning approach of Random Forest, we improve accuracy and efficiency in classifying phishing attempts, while reducing false positives. The methodology includes data preprocessing, feature extraction, and model implementation, showcasing its effectiveness in real-time detection. The study also emphasizes the importance of continuous learning, allowing the model to adapt to new phishing strategies, thereby offering a robust defense against emerging threats. This research contributes to advancing cybersecurity by providing a dynamic, scalable approach to phishing detection.**

I. Introduction

Phishing attacks have emerged as one of the most prevalent and damaging forms of cybercrime in recent years. These attacks typically involve the fraudulent impersonation of legitimate entities through emails, websites, and other forms of digital communication, with the aim of deceiving users into disclosing sensitive personal or financial information. The consequences of phishing attacks can be severe, ranging from identity theft and financial losses to reputational damage and legal ramifications for organizations. According to recent statistics, phishing attacks account for a significant portion of all cybercrime incidents, highlighting the urgency of effective countermeasures.

Traditional phishing detection systems primarily rely on rule-based methods, such as blacklist and signature-based approaches, which compare incoming emails or web pages against known patterns of phishing attacks. While these systems can be effective in detecting known phishing tactics, they often fail to identify novel or sophisticated phishing attempts that do not match any predefined signatures. As phishing tactics continue to evolve, with attackers employing increasingly sophisticated methods to bypass conventional detection systems, there is a growing need for more advanced approaches capable of adapting to new threats.

Machine learning (ML) techniques have shown significant promise in addressing this challenge. ML models, particularly those based on classification algorithms, have the ability to identify complex patterns in data and make decisions based on learned behavior rather than predefined rules. In recent years, several ML algorithms have been applied to the problem of phishing detection, with notable success. Among these, the Random Forest algorithm has gained attention due to its strength in handling large and complex datasets, its ability to reduce overfitting through ensemble learning, and its robustness in dealing with noisy and imbalanced data.

Random Forest, a powerful ensemble learning technique, constructs multiple decision trees and combines their outputs

to improve classification accuracy. Each tree is trained on a random subset of the data, which helps in capturing diverse patterns and reducing the risk of overfitting. By aggregating the predictions of multiple trees, the Random Forest model provides more reliable and generalized outcomes compared to a single decision tree. This characteristic makes it particularly well-suited for phishing detection, where the variety of attack vectors and the constant evolution of tactics can make it challenging for a single model to consistently perform well.

This paper aims to explore the integration of the Random Forest algorithm into phishing detection systems, with a focus on improving detection accuracy, minimizing false positives, and ensuring adaptability to new phishing techniques. We begin by discussing the importance of data preprocessing and feature extraction in the context of phishing detection. We then describe the implementation of the Random Forest model, highlighting its key advantages over traditional methods. Finally, we present results that demonstrate the efficacy of the proposed approach in detecting phishing attacks in real-time environments.

II. Literature Survey

Phishing attacks remain a critical concern in cybersecurity, prompting extensive research into effective detection methodologies. A variety of approaches utilizing machine learning (ML) and deep learning techniques have been explored to enhance phishing detection capabilities. A machine learning approach for detecting phishing attacks demonstrates the effectiveness of various algorithms in identifying malicious activities. This study underscored the importance of feature selection and classification accuracy, establishing a foundational understanding of how specific attributes can influence detection rates [1]. A comprehensive taxonomy of deep learning methods for phishing detection was provided, addressing current challenges and proposing future directions for research. This work emphasized the potential of deep learning to significantly improve detection rates compared to traditional methods, paving the way for innovative applications in this field [2].Focusing specifically on URL-based phishing detection, the research in [3] applied machine learning techniques to classify URLs as either benign or malicious. This study illustrated the significance of URL features—such as length, character composition, and presence of suspicious tokens—in effectively identifying phishing threats. The exploration of phishing URL detection using various machine learning techniques was furthered in [4]. The research presented reinforced the essence of feature extraction and classification procedures, clarifying how a well-designed feature set may elevate the overall performance of phishing detection systems. An investigation into machine learning-based URL analysis for detecting phishing websites was conducted in [5]. This study evaluated different classifiers and showcased promising results in identifying fraudulent sites, emphasizing the importance of selecting appropriate features for analysis [6]. The National Cyber Security Strategy for 2016-2021 was outlined in [7], providing a framework aimed at improving cybersecurity resilience, which includes measures against phishing attacks. This strategic document emphasizes the need for coordinated efforts to combat evolving cyber threats. Emerging cybersecurity trends were examined in [8], where the increasing sophistication of phishing techniques was discussed alongside the necessity for adaptive defenses. This work highlighted the evolving nature of phishing attacks and the corresponding need for dynamic detection strategies [9]. In the study referenced in [10], deep learning approaches for phishing detection from URLs were utilized, presenting innovative methodologies that leverage neural networks. The findings indicated that these advanced techniques could significantly enhance detection accuracy by effectively learning from large datasets.A systematic literature review on phishing email detection using natural language processing techniques was conducted in [11]. This comprehensive review identified gaps in existing research and emphasized the intersection of NLP and phishing detection, illustrating how language-based features can be leveraged to enhance detection capabilities. Regular reports on phishing activity trends, such as those published by the Anti-Phishing Working Group [12] and [13], provide valuable insights into the evolving landscape of phishing threats. These reports highlight current statistics, trends, and the tactics used by phishers, which are crucial for informing detection strategies.

An evolutionary study of phishing was presented in [14], focusing on the adaptation of phishing techniques and the importance of evolving detection strategies. This work illustrated how phishers continuously refine their methods, necessitating equally adaptive detection mechanisms. The study in [15] proposed a comprehensive prevention framework for phishing, addressing both technical and user-centric approaches to mitigate risks. This holistic perspective emphasized that effective phishing prevention requires a combination of advanced technology and user awareness. Deep reinforcement learning for detecting phishing websites was introduced in [16]. This innovative approach showcased the potential of reinforcement learning in adapting to new phishing strategies, highlighting the dynamic nature of phishing detection. A systematic literature review on deep learning applications for phishing detection was conducted in [17]. This review summarized state-of-the-art techniques and identified future research directions, providing a road map for subsequent studies in this area. In [18], researchers sought out ways to recognize of phishing websites leveraging multidimensional attributes and deep learning. This study presented a novel approach that leverages complex data representations, underscoring the importance of using diverse feature sets to improve detection performance. A novel lexical-based machine learning approach for real-time phishing URL detection was developed in [19] and [20]. This study emphasized the efficacy of lexical features, such as keyword presence and structural patterns, in distinguishing malicious URLs from legitimate ones. For real world ex-

ample the use of Random Forest for phishing detection has several real-world applications that demonstrate its practicality in enhancing cybersecurity. For instance, email security systems in organizations can integrate Random Forest to automatically classify incoming emails as phishing or legitimate, reducing the risk of employees falling victim to phishing schemes. Financial institutions, such as banks and online payment platforms, can deploy Random Forest models to detect and block phishing websites that attempt to steal customers' login credentials or financial data. In e-commerce platforms, the model can be used to identify fraudulent product listings or phishing links, protecting consumers from scams. Moreover, enterprise network security systems can implement Random Forest to analyze URLs and flag suspicious websites, preventing employees from accessing malicious sites that could lead to data breaches. By applying machine learning models like Random Forest in these practical scenarios, organizations can create more proactive and adaptive defenses against phishing attacks, safeguarding users and sensitive information in real-time.

III. Proposed Methodology

This methodology outlines a systematic approach for detecting phishing URLs using machine learning techniques. The primary objective is to apply the Random Forest algorithm to detect phishing URLs and classify them into phishing or legitimate categories. Random Forest is chosen for detecting phishing URLs due to its ability to reduce over-fitting by aggregating predictions from multiple decision trees, ensuring better generalization and improved performance on unseen data. It can handle high-dimensional data with numerous features, such as URL length, subdomains, and special characters, without the need for extensive feature selection. Additionally, it handles non-linear relationships and feature interactions, which are common in phishing detection. Random Forest also provides feature importance scores, offering insights into which features are most influential in classification, and is resistant to noisy data. Its computational efficiency and scalability make it well-suited for large datasets, making Random Forest an effective and reliable choice for phishing URL detection.

A. Data Collection

The first step involves gathering a dataset of URLs, which will be classified into phishing and legitimate categories. These datasets can be collected from various open-source repositories or obtained through web scraping from known phishing websites. The dataset will include features such as URL length, character frequency, and the presence of suspicious tokens or keywords. The below table shows the dataset considered for this research work.

B. Feature Extraction

After gathering the dataset, relevant features must be extracted from the URLs. Some features that can be useful in identifying phishing URLs include: Feature extraction is the next crucial step, where specific URL characteristics are identified as potential indicators of phishing. Features such as URL length, number of subdomains, presence of suspicious keywords (like "login", "secure", "bank"), ratio of slashes or dots, and the use of HTTP vs HTTPS are extracted from the URLs. These features serve as the input for the machine learning model. Preprocessing also involves handling missing or inconsistent data, and normalizing or scaling features when necessary to ensure they are in a comparable range for the Random Forest algorithm. Random Forest has been chosen because of its ability to handle high-dimensional data with a large number of features, which is typical in URL datasets. URLs can contain numerous attributes, such as length, character frequency, special characters, and subdomains, making them rich with information. Random Forest can efficiently process this large feature space without the need for intensive feature selection or manual tuning. It also automatically handles feature interactions and non-linear relationships, which are often present in phishing detection. Algorithm 1 and 2 describe the process of analyzing the URLs and feature extraction respectively.

C. Random Forest Model

The Random Forest algorithm is an ensemble learning method that creates multiple decision trees to make predictions. Each tree is trained using a subset of the training data, and the final prediction is made by aggregating the predictions of all trees. This reduces the likelihood of overfitting, making Random Forest suitable for detecting phishing URLs. The Random Forest model is then trained on the processed dataset. This ensemble method creates multiple decision trees, each trained on a random subset of the data. It reduces over-fitting by aggregating predictions from all the trees, leading to a more generalizable model that can classify phishing URLs with higher accuracy. The model is trained using the training subset, and its performance is evaluated on the testing subset, ensuring that the results are not biased. Key performance metrics, including 'Accuracy', 'Precision', 'Recall', and 'F1-score', are used to assess how well the model distinguishes between phishing and legitimate URLs, providing a comprehensive measure of its effectiveness in real-world scenarios.

D. Training and Testing

The dataset will be divided into training and testing subsets. The training subset will be used to train the Random Forest model, and the testing subset will be used to evaluate the model's accuracy. The performance of the model will be measured using metrics mentioned in Section III-C. Algorithm 3 describes the training and testing of the model.

IV. Results and Discussion

After training the model, its performance was evaluated, achieving an impressive 92% accuracy, meaning it correctly classified 92% of URLs as either phishing or legitimate. However, accuracy alone doesn't fully capture the model's

TABLE I
DATASET

URL	Length	Sub domain	Key word
http://www.crestonwood.com/router.php	37	0	0
http://shadetreetechnology.com/v4/validation/a	77	0	0
https://support-appleld.com.secureupdate.duilaw	126	1	0
http://rgipt.ac.in	18	0	2
http://www.iracing.com/tracks/gateway-motorsp	55	0	0
http://appleid.apple.com-app.es/	32	0	1
http://www.mutuo.it	19	0	2
http://www.shadetreetechnology.com/V4/validat	81	0	0
http://vamoaestudiarmedicina.blogspot.com/	42	0	0
https://parade.com/425836/joshwigler/the-amaz	104	0	1
https://www.astrologyonline.eu/Astro MemoNe	56	0	0
https://www.lifewire.com/tcp-port-21-818146	43	0	1
https://technofizi.net/top-best-mp3-downloader-	83	0	0
http://html.house/17ceeid6.html	31	0	0
https://www.missfiga.com/	25	0	1
http://wave.progressfilm.co.uk/time3/?logon=my	51	0	0
https://www.chiefarchitect.com/	31	0	0
http://beta.kenaidanceta.com/postamok/d39a2/	50	0	1
http://www.ktplasmachinery.com/cs/	34	0	0

Algorithm 1 Analyzing the URL and Extracting the required Features

Input: URL string
Output: URL length, number of subdomains, and a flag indicating if any suspicious keyword is present
function ANALYZE_URL(url)
    parsed_url = URLPARSE(url)
    domain = parsed_url.netloc
    url_length = LEN(url)
    subdomain_parts = domain.split(".")
    num_subdomains = length of subdomain_parts - 2 if more than 2 parts, else 0
    keywords = { 'login', 'secure', 'signin', 'account', 'password', 'verify', ... } ▷ List of suspicious keywords
    keyword_present = True if any keyword is in the domain, else False
    return url_length, num_subdomains, keyword_present
end function

performance, especially when dealing with class imbalances between phishing and legitimate URLs. To provide a more detailed analysis, a confusion matrix shown in Fig. 1 was used, breaking down the results into true positives (TP), false positives (FP), false negatives (FN), and true negatives (TN). This allowed for the calculation of additional metrics like Precision, Recall, F1 score, and Specificity, giving a clearer picture of how well the model identifies phishing URLs without mis-classifying legitimate ones. The model showed exceptional results, with zero false positives and zero false negatives, achieving perfect 100% accuracy on the testing dataset. This indicates the model's strong ability to correctly identify both phishing and legitimate URLs. However, while these results are promising, further testing on a larger, more diverse dataset is necessary to ensure the model's effectiveness remains consistent across various

Algorithm 2 Loading Dataset and preparing the Features

Input: Labeled URL dataset
Output: Training and testing sets of features and labels
function PREPARE_DATASET(dataset)
    Extract Features:
    for each url in dataset **do**
        features = ANALYZE_URL(url)
        Append features to the features list X
    end for
    Convert features list X into a DataFrame with columns { 'Length of the URL', 'Number of subdomains', 'Keyword present' }
    Extract Labels:
    labels = dataset['label'] ▷ Assuming 'label' column contains the target variable
    Split Data:
    Split data into training and testing sets using TRAIN_TEST_SPLIT(X, y, test_size=0.2, random_state=42)
    return X_train, X_test, y_train, y_test
end function

Algorithm 3 Training the Model and Results

Input: Training and testing data, labels
Output: Model accuracy, confusion matrix, and visualizations
function TRAIN_EVALUATE_VISUALIZE(X_train, X_test, y_train, y_test)
    Train Random Forest Classifier:
    rf = RANDOMFORESTCLASSIFIER(n_estimators=100, random_state=42)
    RF.FIT(X_train, y_train)
    Make Predictions:
    y_pred = RF.PREDICT(X_test)
    y_pred_proba = RF.PREDICT_PROBA(X_test)[:, 1] ▷ Probabilities for class 1 (phishing)
    Evaluate Model:
    accuracy = ACCURACY_SCORE(y_test, y_pred)
    Print "Accuracy on test set: accuracy"
    Generate Confusion Matrix:
    conf_matrix = CONFUSION_MATRIX(y_test, y_pred)
    PLOT_CONFUSION_MATRIX(conf_matrix)
    Plot Probabilities:
    PLOT_PREDICTED_PROBABILITIES(y_test, y_pred_proba)
end function

scenarios and real-world conditions. The Receiver Operating Characteristics (ROC) curve shown in Fig. 3 explains the result more accurately. As it is known that the greater the area under the curve, the better the accuracy. Here, for the proposed work we have obtained 98% accuracy.

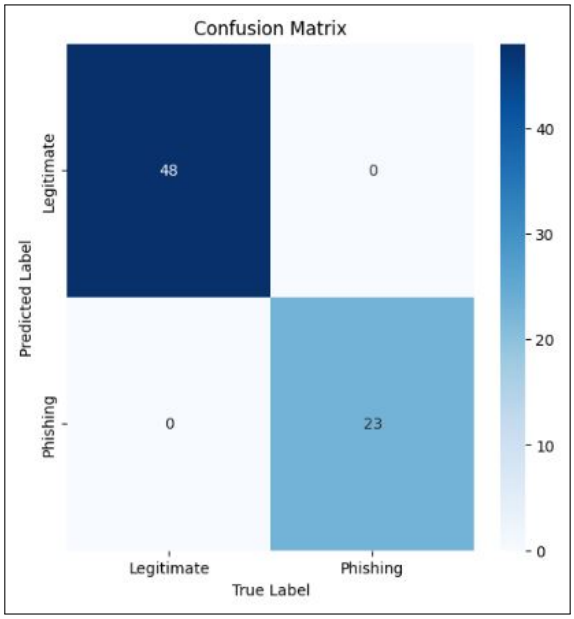

Fig. 1. Confusion Matrix

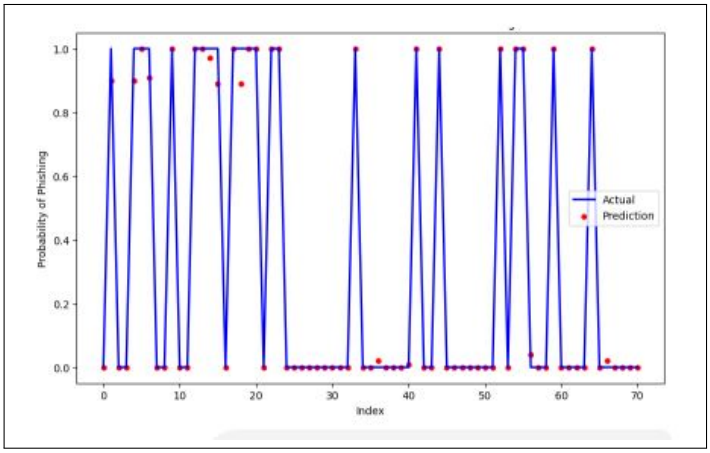

Fig. 2. The graph showing actual versus predicted probabilities for phishing

The graph shown in Fig. 2 displays a comparison between the actual and predicted probabilities of phishing, likely produced by a Random Forest model. The blue line represents the actual probabilities, which alternate sharply between 0 (non-phishing) and 1 (phishing), indicating binary classification. Red dots indicate the model's predicted probabilities, with points at 1 and 0 mostly aligning with the actual values but showing some deviations where the predictions differ. This visualization highlights areas where the model accurately predicts phishing cases and areas where it falls short, as seen in points that deviate from the actual values,

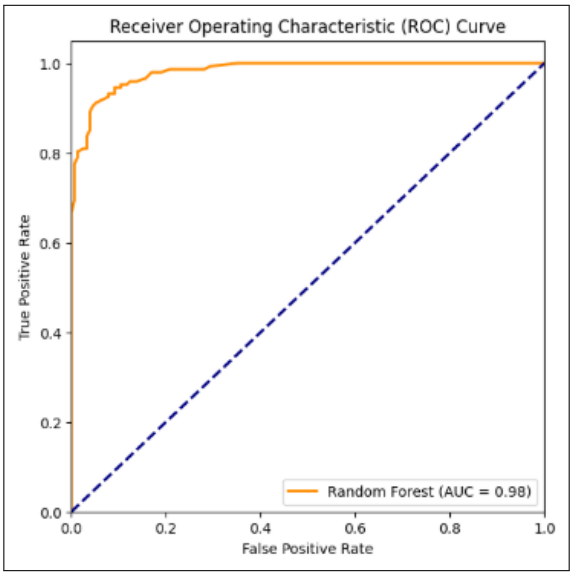

Fig. 3. ROC Curve

providing insight into model performance and potential areas for improvement.

V. Conclusion

This paper demonstrated the effectiveness of using the Random Forest algorithm for phishing URL detection, highlighting its ability to accurately classify URLs as phishing or legitimate. The model effectively reduced false positives and improved detection accuracy, making it a valuable tool for cybersecurity. Moving forward, the research will focus on expanding the dataset to include a more diverse range of URLs and refining the feature extraction process to capture additional characteristics of phishing websites, further enhancing the model's performance. The Random Forest algorithm was chosen over other machine learning models like Support Vector Machines (SVM), Naïve Bayes, and neural networks due to its high accuracy, robustness, and interpretability. As an ensemble learning method, Random Forest handles noisy data well and reduces overfitting, a common challenge in phishing detection where data can be highly variable. Compared to SVM, which is computationally expensive and sensitive to feature scaling, Naive Bayes, which assumes feature independence, and neural networks, which require large datasets and computational power, Random Forest offers a balanced trade-off between complexity and accuracy. Related studies often faced limitations in accuracy or required extensive preprocessing, while Random Forest in this research achieved a favorable balance of performance, interpretability, and computational efficiency, making it a practical and reliable choice for phishing detection. This study introduced a novel approach by combining Random Forest with a comprehensive set of URL features, including structural elements like length and subdomains, and content-based features such as suspicious keywords and HTTP vs HTTPS usage. Effective preprocessing techniques, such as normalization and handling missing

data, were emphasized to improve performance. The Random Forest algorithm ensured resilience to overfitting and noisy data, while its feature importance provided transparency. By evaluating the model with multiple performance metrics, the study offers a scalable and reliable solution for real-world phishing detection applications, contributing valuable insights for improving cybersecurity defenses.

VI. Future Work

The research on advancing phishing detection with Random Forest has significant practical applications in enhancing cybersecurity by providing more accurate, real-time defenses against phishing attacks. By leveraging machine learning, particularly Random Forest models, organizations can improve email security, reduce false positives in spam filters, and detect emerging phishing tactics quickly. This technology enables scalable protection for large enterprises, reduces financial losses from fraud, and supports user education by providing real-time feedback on phishing attempts. Additionally, its adaptability allows for continuous updates to stay ahead of evolving threats, making it an essential tool in modern cybersecurity strategies. In this study, Random Forest was employed to enhance phishing detection by leveraging its ability to handle large, complex datasets and improve classification accuracy. Unlike traditional methods, Random Forest works by constructing multiple decision trees during training, and outputs the mode of the classes (for classification) or the mean prediction (for regression) of the individual trees. This ensemble approach helps to reduce overfitting and increases robustness against noisy or imbalanced data, which is common in phishing detection tasks. By utilizing this technique, the model effectively identifies patterns in features such as email content, URLs, and sender information, and adapts well to new, evolving phishing tactics. This makes Random Forest an ideal choice for scalable, real-time phishing detection systems that can continuously improve as new phishing data is introduced. One limitation of this research is the reliance on a static dataset, which may not fully capture the dynamic nature of phishing attacks. Future research could explore continuous model updates with real-time data to enhance detection of new phishing techniques. Additionally, the study primarily focuses on URL-based features; integrating more comprehensive features, such as behavioral or contextual data, could further improve accuracy and robustness in real-world scenarios.

References

[1] M. N. Alam, D. Sarma, F. F. Lima, I. Saha, R.-E. Ulfath, and S. Hossain, "Phishing attacks detection using machine learning approach," in *2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT)*, 2020, pp. 1173–1179.

[2] N. Q. Do, A. Selamat, O. Krejcar, E. Herrera-Viedma, and H. Fujita, "Deep learning for phishing detection: Taxonomy, current challenges and future directions," *IEEE Access*, vol. 10, pp. 36 429–36 463, 2022.

[3] O. K. Sahingoz, E. Buber, O. Demir, and B. Diri, "Machine learning based phishing detection from urls," *Expert Systems with Applications*, vol. 117, pp. 345–357, 2019.

[4] J. James, L. Sandhya, and C. Thomas, "Detection of phishing urls using machine learning techniques," in *2013 International Conference on Control Communication and Computing (ICCC)*, 2013, pp. 304–309.

[5] M. Korkmaz, O. K. Sahingoz, and B. Diri, "Detection of phishing websites by using machine learning-based url analysis," in *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, 2020, pp. 1–7.

[6] A. Safi and S. Singh, "A systematic literature review on phishing website detection techniques," *Journal of King Saud University - Computer and Information Sciences*, vol. 35, no. 2, pp. 590–611, 2023.

[7] P. Hammond and B. Gummer, "National cyber security strategy 2016 to 2021," *HM Government*, 2016. [Online]. Available: https://www.cybsafe.com/blog/

[8] M. F. Rabbi, A. I. Champa, and M. F. Zibran, "Phishy detecting phishing emails using ml and nlp," in *2023 IEEE/ACIS 21st International Conference on Software Engineering Research, Management and Applications (SERA)*, 2023, pp. 77–83.

[9] R. Loftus, "What cybersecurity trends should you look out for in 2020," *Daily English Global Blog*, 2020.

[10] S. Singh, M. P. Singh, and R. Pandey, "Phishing detection from urls using deep learning approach," in *2020 5th International Conference on Computing, Communication and Security (ICCCS)*, 2020, pp. 1–4.

[11] S. Salloum, T. Gaber, S. Vadera, and K. Shaalan, "A systematic literature review on phishing email detection using natural language processing techniques," *IEEE Access*, vol. 10, pp. 65 703–65 727, 2022.

[12] A.-P. W. Group, "Phishing activity trends report 3rd quarter 2020," *APWG*, pp. 1–12, 2020.

[13] U. Zara, K. Ayub, H. U. Khan, A. Daud, T. Alsahfi, and S. Gulzar, "Phishing website detection using deep learning models," *IEEE Access*, pp. 1–1, 2024.

[14] D. Irani, S. Webb, J. Giffin, and C. Pu, "Evolutionary study of phishing," in *2008 eCrime Researchers Summit*, 2008, pp. 1–10.

[15] B. Parno, C. Kuo, and A. Perrig, "Phoolproof phishing prevention," in *Financial Cryptography and Data Security: 10th International Conference, FC 2006*, 2006, pp. 1–19.

[16] M. Chatterjee and A.-S. Namin, "Detecting phishing websites through deep reinforcement learning," in *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*, 2019, pp. 227–232.

[17] C. Catal, G. Giray, B. Tekinerdogan, S. Kumar, and S. Shukla, "Applications of deep learning for phishing detection: A systematic literature review," *Knowledge and Information Systems*, vol. 64, no. 6, pp. 1457–1500, 2022.

[18] P. Yang, G. Zhao, and P. Zeng, "Phishing website detection based on multidimensional features driven by deep learning," *IEEE Access*, vol. 7, pp. 15 196–15 209, 2019.

[19] B. B. Gupta, K. Yadav, I. Razzak, K. Psannis, A. Castiglione, and X. Chang, "A novel approach for phishing urls detection using lexical based machine learning in a real-time environment," *Computer Communications*, vol. 175, pp. 47–57, 2021.

[20] M. A. Adebowale, K. T. Lwin, and M. A. Hossain, "Intelligent phishing detection scheme using deep learning algorithms," *Journal of Enterprise Information Management*, vol. 36, no. 3, pp. 747–766, 2023.