Phishing Detection using DNS and IP Filtering

Ankita Mandal, Sucheta Chandra, Priyanka Das, Sriparno Chakraborty, Ishika Chowdhury, Kathamrita Ghosh

Dept. of Computer Application and Science,
Institute of Engineering & Management, Kolkata
University of Engineering and Management, Kolkata, India.

*Corresponding author(s). E-mail(s): sucheta.chandra@iem.edu.in;
Contributing authors: ankita.mandal@iem.edu.in;
priyankadas700@gmail.com; Sriparno.Chakraborty2023@iem.edu.in;
Ishika.Chowdhury2023@iem.edu.in; ghoshkathamrita2005@gmail.com;

Abstract

This paper presents a structured approach for detecting phishing URLs by merging data collection, feature extraction, and machine learning techniques. The authors have taken an approach by assembling a diverse dataset of URLs, clearly labeling malicious links as phishing. By leveraging Python's socket and urlparse libraries, we were able to extract DNS records and IP addresses from these phishing links. Key features that were focused on includes IP reputation, types of DNS records, domain age, registration details, and geolocation information. To develop our predictive model, we employed logistic regression model, training it on the features derived from our labeled dataset. The model produces a probability score that indicates how likely a URL is to be malicious. When the model was tested on new URLs, those that exceeded a specific threshold were flagged as phishing. This research enhances phishing detection by effectively combining machine learning with data analysis, allowing us to address the ever-evolving challenges posed by cyber threats.

Keywords: Phishing Detection, Machine Learning, Feature Extraction, Logistic Regression, IP Filtering

1 Introduction

Security is by far the most difficult part of the Internet and communication. There are several ways to compromise internet users' security: the most prevalent is phishing, a type of attack that seeks to steal or misuse a user's personal information, such as account information, accessing personal identities passwords, and credit card numbers. Phishers collect information about users by impersonating trustworthy online resources that are visually indistinguishable. Sensitive information about users may be obtained, putting them at grave risk of financial harm or identity theft. Hence, there is a pressing need to design a system that efficiently detects phishing. Phishing is a cyber attack that deceives individuals into divulging sensitive information, such as passwords and financial details, by using fraudulent emails, text messages, or websites disguised as legitimate sources. Attackers exploit human psychology, often creating a sense of urgency or emotional pressure to prompt immediate action. Common phishing techniques include email phishing, spear phishing, vishing (voice phishing), smishing (SMS phishing), and whaling. Hence, to counter these threats, the authors of this research work has proposed a robust detection system combining DNS and IP address analysis with a Logistic Regression model. This system efficiently identifies suspicious patterns by analyzing features like, domain age, URL length, and keyword presence. It is used for predicting whether a communication is legitimate or a phishing attempt. By continuously learning and adapting to new phishing strategies, the system enhances cybersecurity for both individuals and organizations. Key preventative measures include carefully verifying URLs, ensuring websites use HTTPS, avoiding unsolicited links, and keeping browsers and security software updated. The workflow diagram of the proposed approach is shown in Fig. 1.

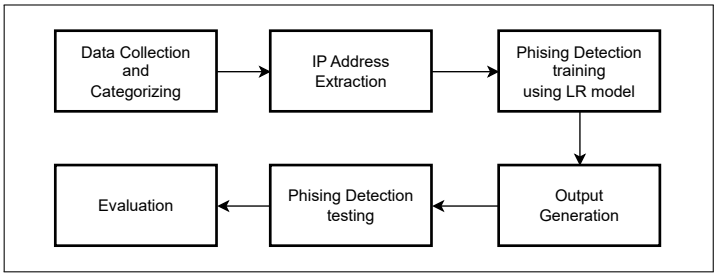


Fig. 1: Block Diagram of the proposed methodology

The remaining part of this research work is organized as follows: Section 2 discusses the existing works in this domain followed by elaborately explaining the methodology in Section 3. Section 4 demonstrates the experimental results and subsequently Section 5 provides the conclusion and probable future works.

2 Literature Survey

Phishing attacks have emerged as a highly sophisticated threat in the digital landscape, driving extensive research aimed at understanding and mitigating these risks. A comprehensive study identified various tactics employed by cybercriminals, underscoring the urgent need for effective countermeasures. This foundational research has paved the way for subsequent studies exploring advanced detection methodologies in [1]. Recent surveys have investigated the application of machine learning techniques in phishing detection, highlighting their potential to significantly enhance security measures. These technologies can effectively reduce the vulnerability of individuals and organizations by analyzing vast datasets and identifying patterns indicative of phishing attempts [2]. Systematic reviews of existing phishing detection methods reveal ongoing advancements, demonstrating that the integration of machine learning applications can lead to improved detection accuracy, thereby providing a robust framework for identifying and neutralizing phishing attacks. Moreover, specialized machine learning approaches tailored for phishing detection underscore the critical role of artificial intelligence in contemporary cybersecurity efforts. In examination of current anti-phishing frameworks has uncovered gaps in existing methodologies, suggesting a need for further exploration and refinement. Such analyses indicate areas for enhancement, paving the way for more effective strategies in combating phishing threats is described in [3]. The development of systems that integrate various machine learning techniques has shown substantial effectiveness in real-world applications, emphasizing the practical implications of these research efforts [4, 5]. Innovative strategies presented at prominent conferences further reinforce the importance of machine learning in phishing detection. The strategies shown in [6] illustrate the ongoing evolution of the field, as researchers strive to develop more sophisticated and adaptive methods for addressing phishing threats mentioned in [5, 7]. Overall, the collective body of research advocates for a multidisciplinary approach to combating phishing, integrating both technological innovations and human factors to strengthen cyber security. As the landscape of cyber threats continues to evolve, ongoing research and the adaptation of strategies will be essential for mitigating the risks posed by phishing attacks [8–10]. The integration of these diverse approaches is crucial for developing comprehensive defenses against this persistent and evolving threat.

3 Proposed Methodology

This section describes our methodical technique of detecting phishing URLs. For example **data collection** phase, a diverse dataset of URLs was gathered from Kaggle, including both legitimate links like https://www.example.com and phishing URLs such as https://secure-login-update.com. Each URL was categorized based on its DNS Records Count (e.g., 3 for `example.com` and 8 for the phishing URL) and labeled for **Phishing Classification** (0 for legitimate, 1 for phishing). During **DNS record and IP address extraction**, a Python script using the `Socket` library resolved the domains and extracted DNS records and IP addresses. For instance, https://www.example.com would return an IP like `93.184.216.34` and multiple DNS records, while the phishing URL https://secure-login-update.com might resolve to

185.24.76.129 with additional A and CNAME records. This information provided a solid foundation for the subsequent logistic regression-based phishing detection model.

3.1 Data Collection

We began by collecting a comprehensive dataset of URLs from Kaggle https://www.kaggle.com/datasets/shashwatwork/web-page-phishing-detection-dataset, encompassing both legitimate and malicious links. This dataset was curated to ensure diversity, allowing for robust analysis. URLs were categorized based on two criteria: DNS Records Count and Phishing Classification. DNS records assess the complexity and potential risk of each URL, with a higher count indicating possible malicious activities. Whereas, for phising classification malicious URLs were labeled as phishing links, facilitating supervised learning in later stages by providing clearly defined examples of both legitimate and phishing URLs. The dataset considered for the proposed work is shown in Table 1

URL	**DNS Record Count**	**Is Phising**
http://www.crestomwood.com/touter.php	0	1
http://shadetreetechnology.com/vt/validation/a111 aedc8ae390eabcfa	0	1
http://rgipt.ac.in	0	0
http://www.iracing.com/tracks/gateway-park/	0	1
http://www.mutuo.it	0	1
http://www.shadetreetechnology.com/V4/validation/ba4b8bddd7958e	1	1
http://vamoaestudiarmedicina.blogspot.com/	0	0
https://parade.com/425836/joshwigler/the-amazing-race-host-phil-keoghan/	0	0
https://www.astrologyonline.eu	0	1
https://www.lifewire.com/tcp-port-21-818146	1	0
https://technofizi.net/mp3-downloader-app-for-android-free-music-download/	0	1
http://html.house/7ceeid6.html	0	0
https://www.missfiga.com/	1	1
http://wave.progressfilm.co.uk/time3/?logon=mypo ste	0	0
https://www.chiefarchitect.com/	0	1
http://www.ktplasmachinery.com/cs/	0	1
http://www.game.co.uk/en/games/nintendo-switch/nintendo-switch/	0	0
https://blog.hubspot.com/marketing/email-open-click-rate-benchmark	0	0

Table 1: Dataset with URL, DNS Record Count, and Phising status

3.2 DNS Record and IP Address Extraction

In this phase, we utilized a Python script to extract DNS records and IP addresses from the collected URLs, employing the Socket library for domain resolution and Urlparse for accurate parsing. Table 2 shows the detailed list. The extraction process addressed scenarios such as domains with port numbers and included exception handling to manage unresolved domains, ensuring dataset integrity. This accurate extraction of DNS records and IP addresses laid a solid foundation for further analysis. The ResolveIPsFromExcel function reads a list of URLs or domain names from

URL	IP Address	DNS Record Count	Phishing
http://www.crestomwood.com/router.php	172.232.25.148	1	1
http://shadetreetechnology.com/Vit/validation/a111aedc8ae3908abcfa	150.60.41.10	0	1
http://rgipt.ac.in	14.139.251.148	0	1
http://www.iracing.com/tracks/gateway-motorsports-park/	54.167.180.60	0	1
http://www.mutuo.it	104.21.2.3	1	1
http://www.shadetreetechnology.com/v4/v58e	15.197.225.128	1	0
http://vamoaestudiarmedicina.blogspot.com/	74.125.26.132	0	1
https://parade.com/425836/joshwigler/the-amazing-race-host-phil-keoghan-previews-the-season-27-premiere/	151.101.194.98	0	1
https://www.astrologyonline.eu	52.49.215.148	0	1
https://www.lifewire.com/tcp-port-21-818146	151.101.2.137	0	1
https://technofizi.net/top-best-mp3-downloader-app-for-android-free-music-download/	104.21.87.203	0	1
http://html.house/7ceeid6.html	104.236.175.252	0	1
https://www.missfiga.com/	23.227.38.74	0	1
http://wave.progressfilm.co.uk/time3/?logon=myposte	162.13.162.62	0	1
https://www.chiefarchitect.com/	54.244.88.174	0	1
http://www.ktplasmachinery.com/cs/	18.222.29.120	0	1
http://www.game.co.uk/en/games/nintendo-switch/nintendo-switch/	23.61.11.178	0	1
https://blog.hubspot.com/marketing/email-open-click-rate-benchmark	104.18.33.148	0	1

Table 2: Dataset with IP Addresses fetched Programmatically

an Excel file, then uses a domain resolution process (e.g., with the Socket library) to retrieve their corresponding IP addresses. It outputs the IP addresses back to the Excel file or stores them for further analysis. The details of the IP address extraction is described in Algorithm 1.

3.3 Phishing Detection Using Logistic Regression

In this step, we employed Logistic Regression model to develop a predictive model for phishing detection. The model was trained using the features extracted from our labeled dataset of URLs. Key aspects of this process included: *Input Features*: The model utilized various features derived from the dataset, including attributes such as DNS records count and IP address characteristics. The input features, such as DNS records count and IP address characteristics, provide valuable information for identifying patterns in domain configurations and network behavior that help distinguish phishing from legitimate URLs.
Target Variable: The target variable was defined as the phishing label (0 for legitimate and 1 for phishing), allowing the model to learn from clearly defined examples.
Model Training: We conducted extensive training sessions on the logistic regression model, adjusting parameters to optimize its performance in distinguishing between legitimate and phishing links. The training process involved evaluating the model's accuracy through cross-validation techniques to ensure reliability.

Algorithm 1 Resolve IP Addresses from URLs

function GetIP(url)
Parse the URL to extract the domain name
Remove any port number from the domain if present
Resolve the domain to an IP address using a domain resolution method (e.g., DNS lookup)
return the resolved IP address
end function
function ResolveIPsFromExcel(input_file, output_file)
Read the input file into a DataFrame named `df`
if 'url' not in df.columns **then**
Raise `ValueError` indicating the missing 'url' column
end if
Initialize an empty list called `data`
for each `url` in the 'url' column of `df` **do**
`ip` ← `GetIP(url)` ▷ Resolve the IP address for the current URL
Append the pair {"URL": `url`, "IP Address": `ip`} to `data`
end for
return `data`
end function
input_file ← 'input1.xlsx' ▷ Specifying the input Excel file
output_file ← 'output.xlsx' ▷ Specifying the output Excel file
`data` ← `ResolveIPsFromExcel(input_file, output_file)` ▷ Call the function to resolve IPs
Create a DataFrame `df` from the `data` list
Print the DataFrame `df` to verify the output

Output Generation: Once trained, the logistic regression model produced a probability score indicating the likelihood of a URL being malicious. This score is pivotal in determining how confidently a URL can be classified as phishing or legitimate.

The logistic regression model is defined by the logistic function $p(y = 1|\mathbf{x}) = \sigma(\mathbf{w}^T\mathbf{x} + b)$, where, $p(y = 1|\mathbf{x})$ is the probability of the instance being phishing, $\sigma(z) = \frac{1}{1+e^{-z}}$ is the sigmoid function,$\mathbf{w}$ is the weight vector, $\mathbf{x}$ is the feature vector, b is the bias term. The loss function used to train the model is the binary cross-entropy loss defined in equation (1). Here, N is the number of samples, $y^{(i)}$ is the actual label for the $i - th$ instance.

$$L(\mathbf{w}, b) = -\frac{1}{N}\sum_{i=1}^{N}\left[y^{(i)}\log(p(y^{(i)} = 1|\mathbf{x}^{(i)})) + (1 - y^{(i)})\log(1 - p(y^{(i)} = 1|\mathbf{x}^{(i)}))\right] \tag{1}$$

Finally, we applied the trained logistic regression model to predict the phishing probability scores for new, unseen URLs. The classification process involved:

Setting a Threshold: A threshold probability score (e.g., 0.75) was established above which URLs would be classified as phishing links. This threshold was determined based on empirical analysis of model performance during training.
Evaluation of New URLs: Each new URL was evaluated against this threshold to ascertain its status as either malicious or legitimate. By applying this rigorous classification method, we aimed to accurately identify potential phishing threats in real-time. This comprehensive methodology effectively integrates data collection, feature extraction (implicitly through logistic regression), and machine learning techniques to enhance phishing detection capabilities and address evolving cyber threats. The details of this approach is described in Algorithm 2.

4 Experimental Results

The phishing detection model, which employed logistic regression, was rigorously evaluated and determined to achieve an accuracy of 0.75. This accuracy signifies that 75% of the test instances were correctly classified, with the model effectively identifying whether URLs were phishing or legitimate. The confusion matrix shown in Fig. 2a provided a detailed breakdown of the model's predictions. Accuracy is defined as $Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$ where, TP =True Positive (correctly identified phishing URLs), TN = True Negative (correctly identified legitimate URLs), FP = False Positive (legitimate URLs incorrectly classified as phishing) and FN = False Negative (phishing URLs misclassified as legitimate) respectively. From the confusion matrix, additional metrics can be derived like $Precision = \frac{TP}{TP+FP}$, $Recall = \frac{TP}{TP+FN}$ and $F1Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$. This distribution offered valuable insights into specific areas where the model performed well and where it struggled. To further assess the model's performance, the ROC curve was plotted as shown in Fig. 2b.The threshold for classifying URLs as phishing or legitimate is set based on the predicted probability score from the logistic regression model. If the predicted probability $P(y = 1|x)$ exceeds the threshold T (e.g., 0.75), the URL is classified as phishing. This threshold is determined empirically by analyzing the model's performance on training data and evaluating metrics like accuracy, precision, and recall.

This curve visually represented the model's ability to discriminate between phishing and legitimate URLs, with the area under the curve (AUC) serving as a quantifiable metric of performance. A higher AUC would indicate a better ability to distinguish between the two classes, reinforcing the effectiveness of the model. Overall, the results suggest that while the model demonstrates a reasonable level of accuracy, there are opportunities for improvement. Potential enhancements could include incorporating additional features, such as URL length, the presence of specific keywords, or patterns associated with phishing attacks. Furthermore, exploring alternative machine learning algorithms or employing ensemble methods could yield better predictive performance. Fine-tuning the logistic regression model through hyper parameter optimization might also enhance its accuracy. Thus, while the current model performs

Algorithm 2 Phishing URL Detection using Logistic Regression

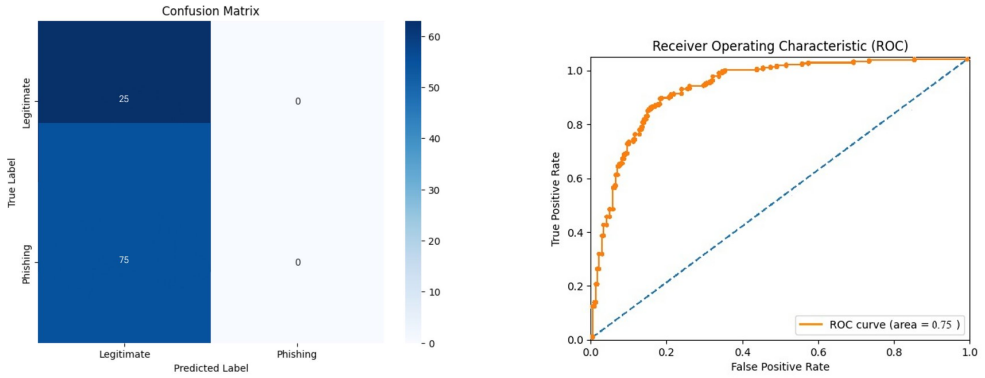
Input: Dataset from Excel file containing URLs, IP addresses, and labels
Output: Accuracy, Confusion Matrix, ROC Curve
procedure PHISHINGDETECTION
    Load dataset ← `urls_dataset5.xlsx`
    Print initial shape and first few rows of the DataFrame
    Rename columns to {url, ip_address, label, is_phishing}
    Drop rows with missing values

    function CLEAN_IP(ip)
        if is valid IP(ip) **then**
            ip ← `IPv4Address(ip)`
            return `int(ip)`
        else
            return None
        end if
    end function
    Apply `clean_ip` to `ip_address` column
    Drop rows with invalid IPs
    Print cleaned DataFrame

    Define features X ← `ip_address` and target y ← `is_phishing`
    if X is empty **then**
        Raise ValueError("Feature set is empty")
    end if
    Scale features X using `StandardScaler()`
    Split data into training and testing sets: $X_{train}, X_{test}, y_{train}, y_{test}$
    Print shapes of training and testing sets
    Train Logistic Regression model on X_{train}, y_{train}
    Evaluate model y_{pred} ← `predict`(X_{test})
    Calculate accuracy ← `accuracy_score`(y_{test}, y_{pred})
    Print accuracy, predictions, and confusion matrix

    function PLOT_CONFUSION_MATRIX(cm, classes)
        Plot confusion matrix using `seaborn`
    end function
    Generate and plot confusion matrix

    if len$(y_{test}) > 0$ and len$(X_{test}) > 0$ **then**
        Compute ROC curve and AUC
        Plot ROC curve
    else
        Print "Not enough data for ROC curve plotting."
    end if
end procedure



(a) confusion matrix (b) ROC curve

Fig. 2: Confusion Matrix and Receiver Operating Characteristics Curve

satisfactorily, ongoing efforts to refine and improve its capabilities are warranted to achieve even greater efficacy in phishing detection. This research emphasizes the need for a systematic approach to phishing URL detection amidst evolving cyber threats. We combined data collection, DNS record and IP address extraction, and machine learning techniques to develop a robust detection algorithm. The dataset from Kaggle, including both legitimate and phishing URLs, ensured diverse training data. Feature extraction highlighted key attributes, such as IP address reputation and domain age, for risk evaluation. Logistic regression effectively classified URLs, with probability scores aiding decision-making.

5 Conclusion and Future Works

In this research paper, we have outlined a thorough method for detecting phishing URLs through a systematic algorithm that integrates data collection, feature extraction, and machine learning techniques. By gathering a varied dataset of URLs from Kaggle, which includes both legitimate and malicious links marked as phishing, we laid a strong groundwork for our analysis. We utilized Python's socket and urlparse libraries to extract DNS records and IP addresses, which provided essential information about the URLs. The code was designed to effectively manage scenarios involving port numbers and cases where domain names could not be resolved, thereby maintaining the integrity of our dataset. During feature extraction, we pinpointed important characteristics from the DNS records and IP addresses, including IP address reputation, DNS record types, domain age, registration details, and geolocation data. These features were crucial in evaluating the potential risk associated with each URL. The logistic regression model formed the core of our phishing detection system. By training the model on the labeled dataset with the extracted features, we were able to produce probability scores that indicated the likelihood of a URL being malicious. The model's effectiveness in accurately differentiating between legitimate and phishing links was essential in tackling the ever-evolving threat of phishing attacks. Ultimately, we applied the trained logistic regression model to new, unseen URLs, classifying those

with probability scores exceeding a specific threshold (e.g., 0.75) as phishing links. This step showcased the practical application of our algorithm in real-world situations, facilitating the identification and mitigation of potential phishing threats. Although The methodology discussed in this paper has the potential for further refinement and enhancement.The importance of this research lies in its contribution to the field of cybersecurity. By merging machine learning techniques with traditional data analysis methods, we have created an effective strategy to combat phishing attacks.

References

[1] Alhassan, M., *et al.*: Study on phishing attacks. International Journal of Computer Applications **182**(33), 27–29 (2018)

[2] Mahmood, W., *et al.*: Phishing attack detection using machine learning. In: 2021 IEEE International Conference on Cyber Security and Cloud Computing (CSCloud), pp. 163–168 (2021)

[3] Hussain, Z., *et al.*: Anti-phishing framework: A survey. Software: Practice and Experience **2**, 1208–1211 (2018)

[4] Riaz, A., Khan, A.: A systematic review of phishing detection. Expert Systems with Applications **230**, 119886 (2023)

[5] Khan, M., *et al.*: A novel approach to phishing detection. Journal of Ambient Intelligence and Humanized Computing **10**, 3233–3242 (2018)

[6] Ahmed, M., *et al.*: Phishing detection using machine learning: A survey. Security and Privacy **1**(1), 1–18 (2017)

[7] Li, W., Manickam, S., Laghari, S.U.A., Chong, Y.-W.: Uncovering the cloak: A systematic review of techniques used to conceal phishing websites. IEEE Access **11**, 71925–71939 (2023)

[8] Mao, J., Tian, W., Li, P., Wei, T., Liang, Z.: Phishing-alarm: Robust and efficient phishing detection via page component similarity. IEEE Access **5**, 17020–17030 (2017)

[9] Zhang, J., *et al.*: A phishing detection system based on machine learning techniques. Applied Soft Computing **70**, 682–696 (2018)

[10] Dhamija, R., Tygar, D.: Phishing detection: A machine learning perspective. In: Proceedings of the 15th ACM Symposium on Information, Computer and Communications Security, pp. 411–416 (2022)