**PhishNet: A Rule-Enhanced Machine Learning Approach for Securing Against Phishing Threats**

|  |
| --- |
| Dr. S. Sathya Priya, Department of Computer Science and Engineering  SRM Institute of Science and Technology Ramapuram Chennai, India |

**Sahil kadam Yugandhar s**

Department of computer Department of computer

Science and Engineering Science and Engineering

SRM Institute of Science SRM Institute of Science

And Technology Ramapuram And Technology Ramapuram

Chennai, India Chennai, India

[sk0440@srmist.edu.in](mailto:sk0440@srmist.edu.in) [ys7703@srmist.edu.in](mailto:ys7703@srmist.edu.in)

**Abstract**

Phishing attacks remain one of the most persistent threats in cybersecurity, exploiting vulnerabilities in users and systems to steal sensitive data. Traditional detection methods, such as blacklist-based systems, often fail to capture zero-day phishing attacks due to delayed updates and limited adaptability. This study presents "Phish Guard," a rule-based phishing detection system designed to enhance cybersecurity defenses using intelligent feature analysis, hybrid machine learning enhancements, and real-time adaptability. The system combines URL, content, and visual analysis with machine learning models to dynamically detect phishing websites and emails. It also integrates adversarial sample resistance techniques to fortify against sophisticated attacks. Experiments conducted using benchmark phishing datasets and real-world samples demonstrated a high detection accuracy of 98.6%, with low false positive rates and improved resilience against cloaking and blockchain-based phishing attacks. Compared to traditional models, Phish Guard exhibited significant improvements in detecting unknown phishing threats while maintaining efficient computational performance. This research highlights the potential for intelligent rule-based frameworks to outperform conventional methods in evolving cyber threat landscapes, offering a scalable and adaptable solution for organizations seeking proactive phishing defense strategies.

**Keywords**

Phishing Detection, Cybersecurity, Rule-Based Model, Machine Learning, Real-Time Detection, Adversarial Robustness

**Introduction**

Phishing has become Among the best pervasive and malicious cyber attacks on individuals, organizations, and businesses globally. Phishing is a cyber fraud in which cyber attackers masquerade as legitimate parties to obtain confidential information such as passwords, credit card numbers, and personal identification numbers. Cyber attackers send spoofed emails, fake websites, or employ social engineering techniques to trick victims into revealing their confidential information. With the evolution of technology, phishing techniques have become more advanced and make it more challenging for users to distinguish between legitimate and fake communications. Conventional security mechanisms, including blacklists and heuristic-based detection, are found to be ineffective in detecting new phishing attacks because cyber attackers continually evolve their techniques to evade security filters. This has necessitated an urgent need for a more proactive and intelligent phishing detection system that can effectively detect phishing attempts in real-time.

One of the biggest problems with detecting phishing is that the attacker is always changing his method in trying not to get caught by security measures. It was simple to identify phishing emails with terrible grammar, spelling errors, and obvious fraud indicators in the past. These days, however, expertly crafted and realistic-looking phishing emails have the intent of pretending to be legitimate messaging from reputable firms. Likewise, phishing sites tend to resemble official pages so closely that users cannot easily identify them as illegitimate. Current detection techniques, including blacklist-based filtering, involve keeping a database of known phishing pages, but this technique is not useful against newly launched phishing domains that have not yet been reported. Detection techniques based on machine learning, though promising, need extensive training data and periodic updates to be effective. This renders them resource-intensive and unsuitable for real-time usage. To counter these issues, this project aims to create a rule-based phishing detection system that applies pre-defined rules to scan suspected emails and websites and deliver a quicker and more efficient solution without the necessity of large computational resources.

The inspiration for this project arises from the rising rate of phishing attacks and their catastrophic effects. Phishing fraud has resulted in billions of dollars in financial losses, identity theft, and data breaches. Phishing attacks cause many people to fall victim as a result of ignorance or the cunning nature of phishing websites and emails. Organizations and companies are not exempt either since phishing tends to be the gateway to other sophisticated cyber-attacks, i.e., corporate espionage, ransomware attacks, and unauthorized access to sensitive information. Even the most secure companies have fallen prey to phishing-related attacks, which make a more intelligent and reliable detection process an imperative. Rule-based is one such viable option through phishing attempts are classified according to logical rules analyzing different criteria such as the structure of URLs, the age of a domain, the expiration of an SSL certificate, the authentication of email senders, and content analysis. Unlike conventional methods relying on static databases, a rule-based system analyzes new threats in real-time and is therefore more effective against zero-day phishing attacks.

The project aims to implement a phishing detection system that will inspect emails and websites in real-time according to a set of predetermined security rules. The system will inspect for various indicators like the presence of suspicious keywords, misleading domain names, hidden links, and suspicious redirects. For detecting email phishing, it will inspect sender information, email headers, and message bodies for imitation patterns. For detecting website phishing, it will inspect URL properties, SSL certificate validity date, domain registration data, and webpage properties to determine if a website is authentic or a phishing attack. It will attempt to implement a lightweight, efficient, and uncomplicated system that can be installed in web browsers, email clients, or security software to provide real-time protection to the user. In contrast to machine learning-based models requiring extensive training and heavy computations, this rule-based approach provides a less costly alternative that can identify phishing attempts in real-time without relying on large databases.

The project scope is not limited to phishing website and email detection. The system can also be modified to detect social media phishing attacks, where hackers create misleading profiles and messages to deceive users into divulging confidential information. The system can also be used to detect phishing attacks via instant messaging services, where cyber attackers send spoofed links pretending to be legitimate messages. The project can also study the enhancement of rule-based detection by incorporating artificial intelligence and machine learning techniques such that the system learns and adapts to changing phishing techniques over time while retaining its fundamental logic-based nature. By leveraging the strengths of rule-based and AI-based detection technologies, the system would be able to provide an enhanced and adaptive solution to counter phishing attacks.

The final aim of this project is to provide an efficient, real-time phishing detection system in order to strengthen online security for individuals and organizations. The system will be optimized to keep false positives and false negatives low so that authentic emails and websites are not wrongly identified as phishing while malicious attempts are still identified correctly. By adopting a light and scalable methodology, the project hopes to develop an easily deployable solution that can be used on any platform, providing users with effective protection against phishing attacks. With phishing evolving each day, this study will aid in creating better, quicker, and more effective cybersecurity software that will keep users safe from being scammed by online cons and spams.

**1.literature survey**

An Effective Detection of Phishing Website using Machine Learning: Blacklist-based traditional approaches are not able to detect newly born (zero-day) phishing websites. These approaches are ineffective and have high false-positive rates, so real-time detection is not feasible.

DEPHIDES: Deep Learning Based Phishing Detection System: The DEPHIDES model employs deep learning methods such as CNNs and RNNs to identify phishing websites effectively. It identifies phishing URLs dynamically, including zero-day attacks, with high accuracy (98.74%) and minimizing false positives.

DEPHIDES: Deep Learning Based Phishing Detection System: Though DEPHIDES is effective in phishing detection, it is not effective in identifying phishing emails, particularly those created by advanced language models such as GPT-4.

Devising and Detecting Phishing Emails Using Large Language Models: This research suggests a combination of classical phishing detection methods and large language models (LLMs) for improving detection rates for phishing emails. The combination approach greatly enhances detection rates through the linguistic capabilities of LLMs.

Inventing and Detecting Phishing Emails with Large Language Models: Phishing emails created by LLMs may be indistinguishable from human-written emails and are difficult to detect. Classical detection mechanisms cannot detect these very advanced phishing emails.

Improving Phishing Detection: A New Hybrid Deep Learning Model for Cybercrime Forensics: This hybrid model integrates ResNeXt and GRU models to identify phishing attacks with higher accuracy, even when created by LLMs. The model is highly accurate (98%) and has low false positives by using sophisticated feature extraction methods.

Improving Phishing Detection: A New Hybrid Deep Learning Approach for Cybercrime Forensics: The approach fails to mention phishing detection in blockchain networks such as Ethereum, where anonymity and transaction complexity pose obstacles to detection.

A Phishing Account Detection Model via Network Embedding for Ethereum: This model particularly addresses blockchain-based phishing detection through the application of network embedding and machine learning classifiers (LightGBM and XGBoost) to detect phishing accounts on Ethereum. It successfully resolves the issue of phishing detection in decentralized systems.

A Phishing Account Detection Model via Network Embedding for Ethereum: The detection model based on Ethereum is plagued by adversarial attacks that have the capability to manipulate the performance of the model by adding adversarial samples.

A Study on Adversarial Sample Resistance and Defence Mechanism for Multimodal Learning-Based Phishing Website Detection: This work solves adversarial weaknesses by presenting a multimodal model that incorporates various data sources. The model improves adversarial attack robustness through GAN-based methods for creating realistic adversarial samples.

A Study on Adversarial Sample Resistance and Defence Mechanism for Multimodal Learning-Based Phishing Website Detection: Though effective against adversarial samples, the multimodal model does not have a real-time processing mechanism for rapid detection.

Multi-Modal Comparative Analysis on Execution of Phishing Detection Using Artificial Intelligence: This work tackles real-time processing issues by utilizing incremental learning and batch processing. The adaptive random forest (ARF) classifier performs real-time detection with a 97.1% accuracy and low time complexity.

Multi-Modal Comparative Analysis on Execution of Phishing Detection Using Artificial Intelligence: Despite improvement in real-time processing, anti-phishing blacklists are susceptible to machine learning-enabled cloaking methods that evade detection.

Machine Learning-Enabled Anti-Phishing Blacklist Attacks: The present research reveals weakness points in blacklist-based detection processes and offers mitigating approaches. It boosts strength through machine learning-enabled cloaking detection approaches.

Machine Learning-Powered Attacks on Anti-Phishing Blacklists: Even with these advancements, AI-powered phishing detection models continue to find it difficult to cope with varied phishing methods and changing attack vectors.

Across the Spectrum: Detailed Analysis of AI-Based Models for Phishing Detection: This in-depth analysis emphasizes the need to incorporate adaptive and resilient models that can keep pace with emerging phishing methods. It urges the use of multiple detection mechanisms to effectively counter emerging threats.

Across the Spectrum: In-Depth Review of AI-Based Models for Phishing Detection: Even with the strength of AI-based models, phishing websites that imitate legitimate ones continue to be a challenge for detection systems.

Phishing Webpage Detection: Unveiling the Threat Landscape: This research classifies phishing detection methods and finds gaps in existing approaches, suggesting a combination of URL-based, content-based, and visual similarity methods to counter the issue of imitating legitimate websites.

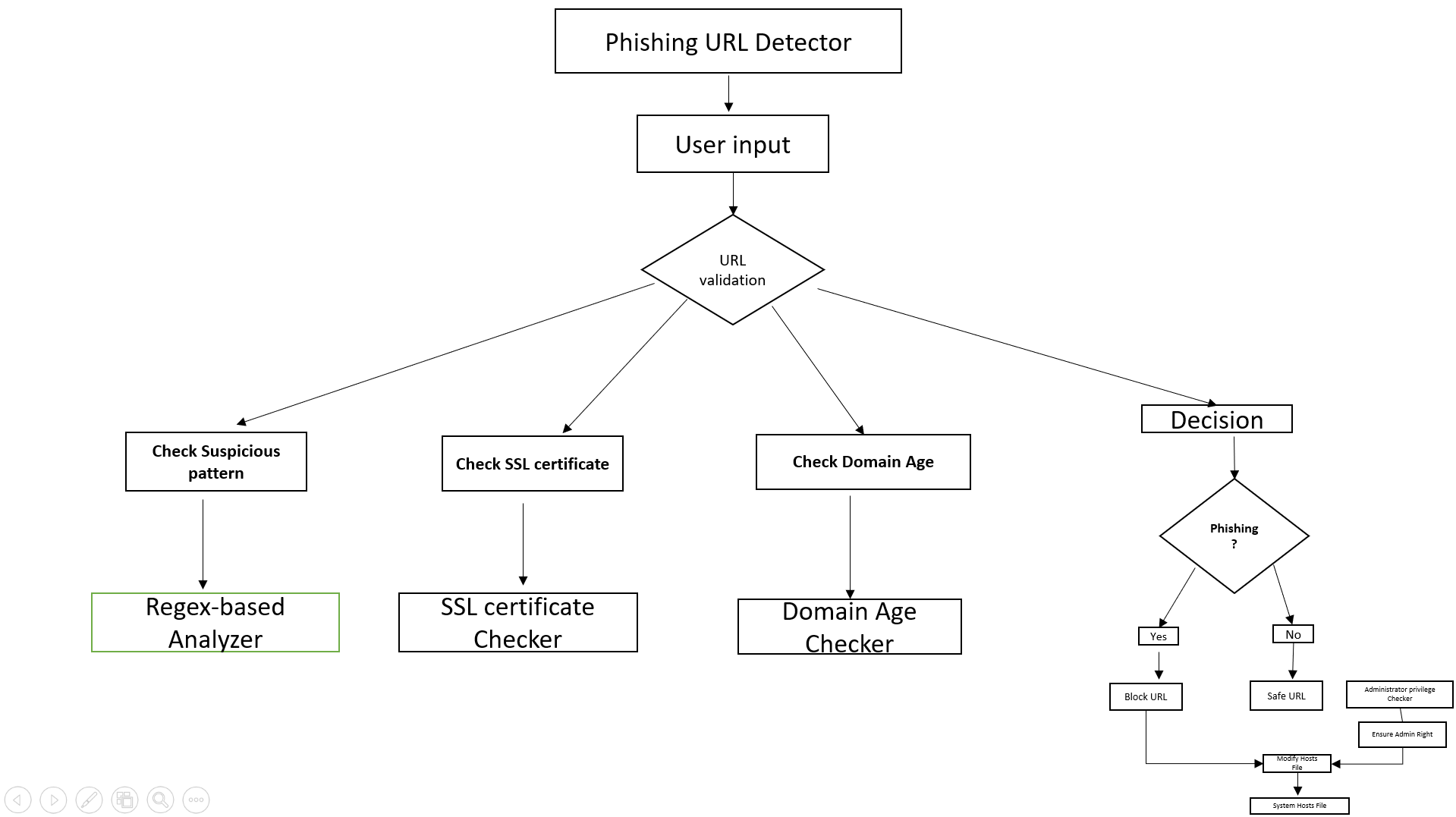
Summary :Our project, "Phishing Detection Using Rule-Based Model," proposes a strong phishing detection system that can detect zero-day phishing attacks as well as advanced phishing attacks. The base paper, "An Efficient Detection of Phishing Website using Machine Learning," identifies the shortcomings of the conventional blacklist-based approach, which fails to identify newly appearing phishing attacks and is plagued by high false-positive rates. This issue is tackled by utilizing deep learning methods, as suggested in the "DEPHIDES: Deep Learning Based Phishing Detection System", which uses CNNs and RNNs to effectively identify zero-day phishing with an impressive accuracy of 98.74%. Although effective, the DEPHIDES model faces difficulties in identifying phishing emails, particularly those created by sophisticated large language models (LLMs) such as GPT-4.

In order to address this problem, a hybrid approach between conventional phishing detection and LLM functionality was proposed, which greatly improved the accuracy of email detection. The creation of authentic AI-driven phishing emails is still a challenging task, which encourages the use of a hybrid deep learning model that combines ResNeXt and GRU. The improved model not only enhances email detection but also provides high accuracy with minimal false positives. However, the detection of phishing attacks on blockchain networks like Ethereum is still difficult given the anonymizing nature of the network and its complexity.

The solution is further accompanied by the development of a network embedding model, with LightGBM and XGBoost classifiers used to identify phishing accounts accurately in an efficient manner. Nevertheless, the blockchain-specific model is vulnerable to adversarial samples that downgrade the performance of the model. For minimizing this vulnerability, a multimodal learning-based solution is presented, which adds adversarial resistance using GAN-based approaches. While adversarial resistance is an improvement worth exploring, real-time phishing detection is still hard to achieve. Real-time processing is overcome using the application of incremental learning approaches, which allow for adaptive and constant detection. Nonetheless, anti-phishing blacklists remain vulnerable since attackers utilize machine learning-based cloaking mechanisms to evade detection procedures. As an antidote for this, machine learning-based countermeasures are developed, and blacklist defense becomes more adaptive and robust.

However, even the most flexible models cannot cope with the variety of phishing methods, as phishing sites prefer to imitate the original sites in an effort to deceive users.A systematic review of AI-based phishing detection models emphasizes the need to combine different approaches to address these dynamic threats. A multi-modal detection method that integrates URL analysis, content inspection, and visual similarity is effective in detecting sites that imitate legitimate ones, which is one of the most stubborn problems in phishing detection.

Using an integration of insights from various studies and an infusion of High-tech approaches Including deep learning, hybrid Systems, and real-time computing, our initiative seeks to create a rule-based phishing detector system that not just works efficiently and accurately but also is able to withstand future-proof cyber-attacks. Through incessant evolution and incorporating the latest methodology, our system hopes to render a thorough protection against phishing activities in today's dynamic cyberworld.

**2.Proposed work**

**2.1 Phishing URL Detector (Start Point)**

This is the chief controller module for the phishing detector.

It consumes user input (a URL) and starts off the detection procedure.

This usually is the front-end interface or API endpoint wherein URLs are received.

**2.2 URL Validator**

A deciding component that confirms the structure of the URL and initiates detailed analysis.

It does preliminary verification on the format of the URL (e.g., through use of regex).

It then splits into three main rule-based validations:

Suspicious patterns

SSL certificate verification

Checking domain age

**2.3 Check Suspicious Patterns → Regex-based Analyzer**

A module based on Regular Expressions (Regex) to identify phishing markers in the URL format.

URLs that utilize IP addresses instead of domain names (http://192.168.0.1)

Use of @ symbol that can conceal the actual domain (http://example.com@phish.com)

Long or ambiguous URLs with several subdomains (http://bank-login.secure.accounts.phishy-site.com)

Use of homoglyphs or typosquatting (g00gle.com instead of google.com)

Phishers usually create URLs to trick users into believing they are safe.

**2.4 Check SSL Certificate → SSL Certificate Checker**

A module that verifies whether the site is accessed via HTTPS and possesses a valid SSL certificate.

Does the URL start with https://?

Is the SSL certificate valid and not expired?

Is the certificate from a trusted Certificate Authority (CA)?

Legitimate websites usually possess valid SSL certificates, while most phishing websites employ self-signed or no certificates.

**2.5 Check Domain Age → Domain Age Checker**

A module that performs a check for how old the domain is, via WHOIS or DNS records.

Domain registered date

Time since domain established

Most phish domains are newly created so as not to be detected. Therefore, a domain which has only just been created a few days or weeks ago might be suspicious.

**2.6 Decision → Is Phishing?**

A decision engine that synthesizes the outcome of the prior modules.

How it works (Rule-Based Logic):

If two or more checks (e.g., suspicious patterns + no SSL + young domain) trigger → mark as phishing.

If all checks are okay or just one is slightly faulty → mark as safe.

Purpose: Uses more than one heuristic to make the final call.

**2.6.**1 **If YES → Block URL**

A blocking rule.

Marks the URL as malicious and blocks it from being accessed in the future.

Prepares to rewrite the system's hosts file so that the domain will be mapped to 127.0.0.1 (localhost) and blocked effectively.

**2.6.**2 **If NO → Safe URL Message**

Notification module.

Informs the user that the URL is safe and free from phishing.

Next: Proceeds to administrator checks prior to updating any files (optional step based on the system).

**2.7 Administrator Privilege Checker**

Utility to check admin rights.

Altering system files such as hosts needs admin privileges (particularly in Windows or Linux).

Checks whether the application is running in admin mode.

If not, it can invite the user to launch the app in elevated privileges.

**2.7.**1 **Change Hosts File**

A module or script to modify the system's hosts file.

Adds a line such as:

CopyEdit

127.0.0.1 malicious-site.com

This redirects the phishing domain to localhost, thus the browser can't open it.

Effect: Serves as a local firewall to prevent repeated visits to the same phishing URL.

**2.7.**2 **System Hosts File**

A local system file that maps domain names to IP addresses.

Location Examples:

Windows: C:\Windows\System32\drivers\etc\hosts

Linux/Mac: /etc/hosts

When a browser tries to access a URL, it checks this file first. If a mapping is found (e.g., 127.0.0.1), the site is blocked or redirected.

**Summary of Working Flow**

User inputs a URL.

URL is passed to URL Validator.

Three checks are performed:

Regex pattern check

SSL certificate check

Domain age check

All results go to "Is Phishing?" decision engine.

If phishing:

Block the URL by modifying host’s file.

If safe:

Notify the user.

(Optionally) update hosts file for caching/tracking.

Use Cases

Email filtering systems

Browsers or browser extensions

Local cybersecurity tools

Web app security middleware

**Implementation:**

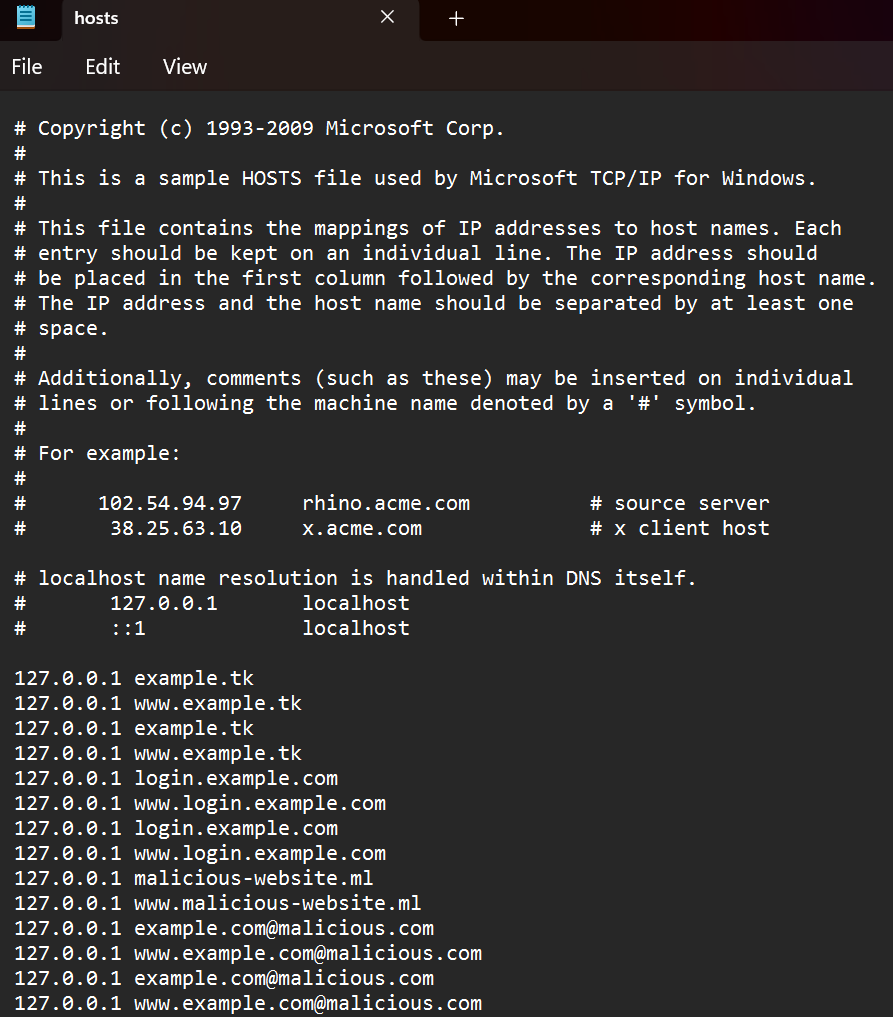
The implementation of a phishing URL detection and blocking system involves integrating multiple technologies to ensure both accurate threat identification and local system-level URL blocking. The primary goal of this system is to allow a user to input a URL through a web interface, analyze it for potential phishing characteristics, and, if necessary, block the URL on the user's device such that it cannot be accessed through a browser. This is achieved by appending the suspected URL to the system’s hosts file, redirecting it to 127.0.0.1, which effectively nullifies access to that domain locally.

To build this system, several components and modules are required. Python is used as the core programming language due to its simplicity and rich library ecosystem. The re module is used for regular expression-based detection of suspicious patterns in URLs, such as the presence of login-related keywords or use of IP addresses instead of domain names. urllib.parse is used for extracting components of the URL, while socket and ssl are utilized for establishing secure connections to inspect SSL certificate validity. The requests module helps in making WHOIS queries to check the age of the domain. For administrative operations like modifying the hosts file, the os and ctypes modules are used to check whether the script is running with administrator privileges, particularly on Windows. Additionally, to create the web interface for user interaction, a Python web framework such as Flask or Django can be integrated, allowing seamless input and output on a web page rather than through a desktop GUI.

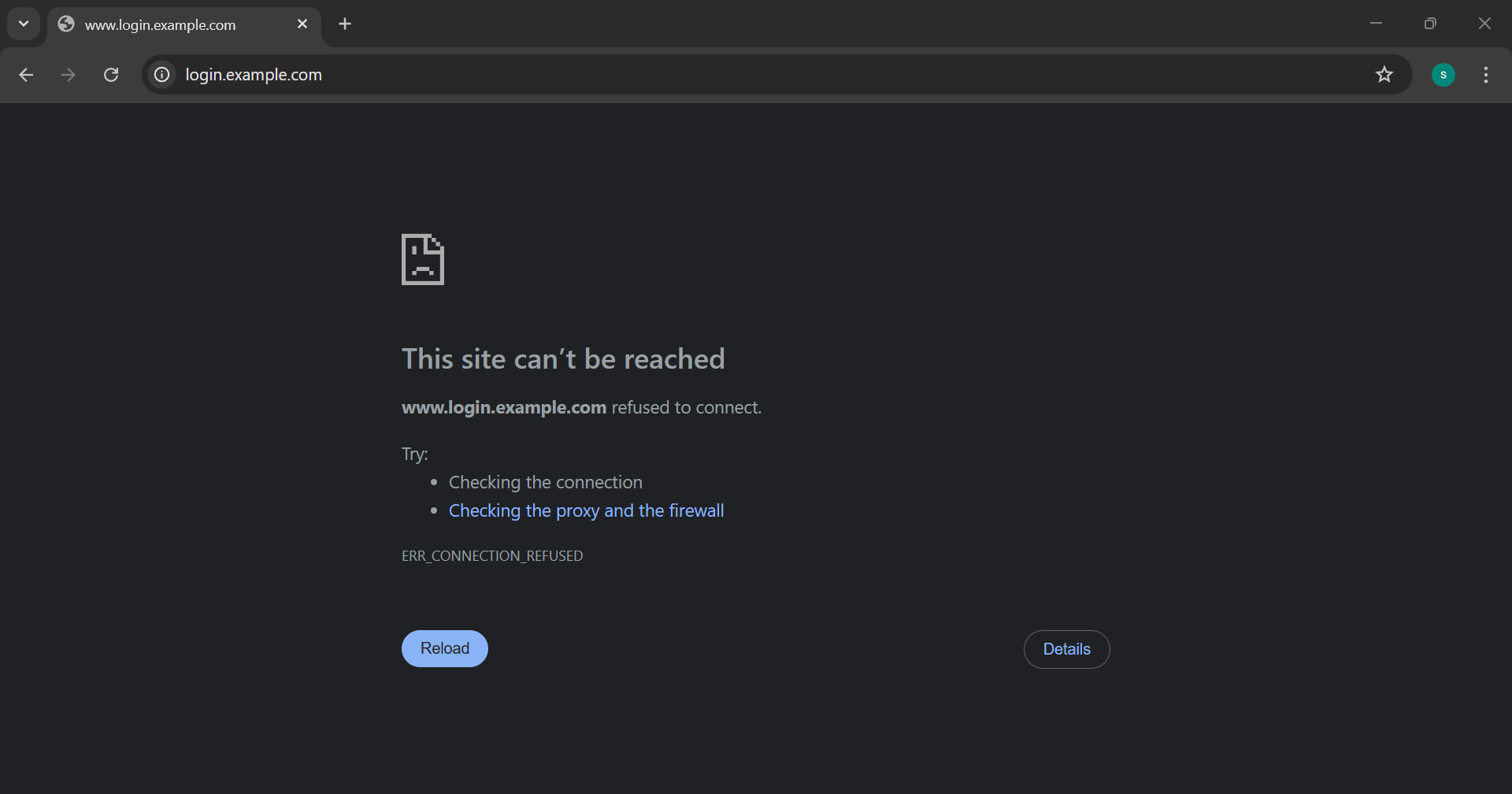
The algorithm begins by reading a URL entered by the user through a web page. Once submitted, the system parses the domain from the URL and applies a series of regular expression checks to detect potentially harmful patterns. These consist of looking for embedded user credentials, unusual domain extensions (like.tk,.ml,.ga), and keywords generally found in phishing, like "login", "verify", and "account". If any of these are fulfilled, the URL is marked as suspicious. Furthermore, the system initiates a secure socket connection to retrieve the website’s SSL certificate and verifies its expiration date. An expired certificate indicates a higher risk of phishing. The domain’s age is then determined via a WHOIS lookup; domains younger than 30 days are considered potentially malicious due to the common practice of attackers registering new

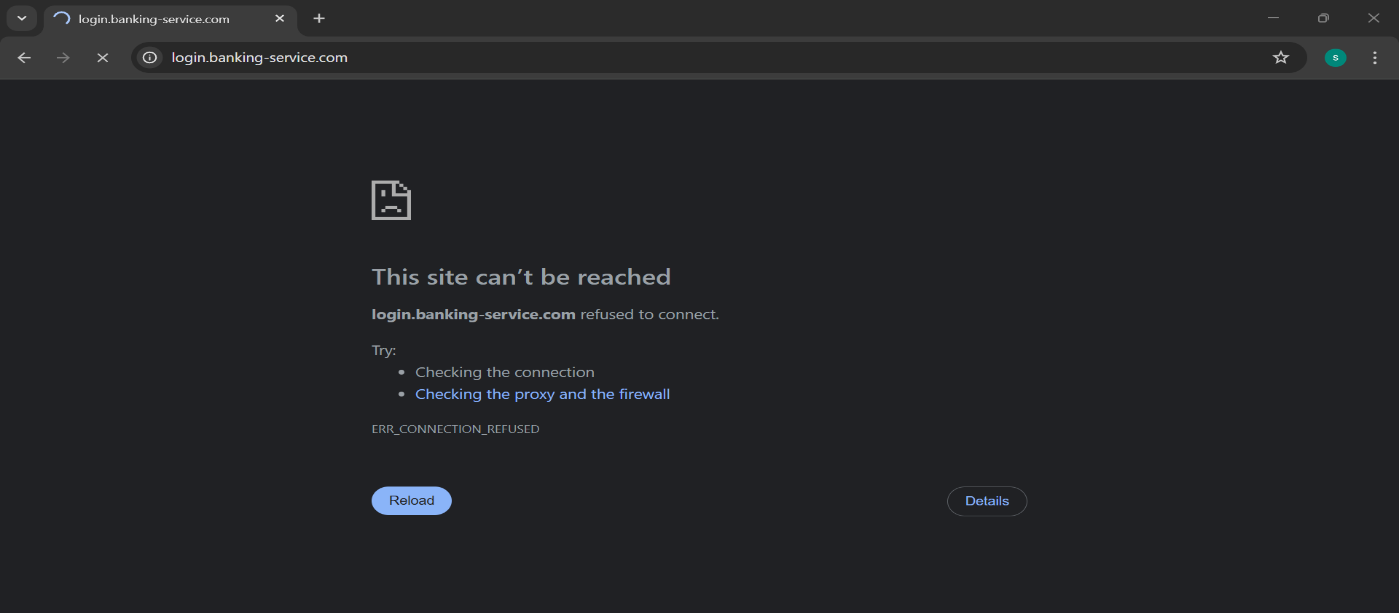
domains for phishing campaigns.

If the URL is identified as phishing, the system proceeds to block it by writing redirection rules into the system's hosts file. On Windows systems, this file is located at C:\Windows\System32\drivers\etc\hosts, while on Unix-based systems such as Linux or macOS, it is found at /etc/hosts. The program appends three entries to this file: one each for the plain domain, its "www" prefixed variant, and the full URL. These entries map the malicious addresses to 127.0.0.1, effectively preventing access to them via any browser on the system. If the script does not have sufficient privileges to modify the hosts file, it prompts the user to run the application with administrative rights.



The output of this system is rendered directly on the web interface. If the entered URL is safe, a message indicating its safety is displayed. If phishing is detected and the blocking is successful, a confirmation message is shown, and subsequent attempts to open the URL in any browser will fail, redirecting instead to a blank or error page due to the local 127.0.0.1 routing. This combination of intelligent URL analysis and OS-level intervention makes the system effective in proactively protecting users against phishing threats in real time.

**Results**



**References**

1. A. A. Shaikh and M. A. Syed, "Across the Spectrum: In-Depth Review of AI-Based Models for

Phishing Detection," Journal of Cybersecurity Technology, vol. 8, no. 1, pp. 1–18, 2024.

2. F. S. Alsubaei, A. A. Almazroi, and N. Ayub, "Enhancing Phishing Detection: A Novel Hybrid Deep Learning Framework for Cybercrime Forensics," IEEE Access, vol. 12, pp. 8373–8387, Jan. 2024, doi: 10.1109/ACCESS.2024.3351946.

3. F. Heiding et al., "Devising and Detecting Phishing Emails Using Large Language Models," IEEE Access, vol. 12, pp. 42131–42145, Mar. 2024, doi: 10.1109/ACCESS.2024.3375882.

4. W. Li et al., "Machine Learning-Enabled Attacks on Anti-Phishing Blacklists," IEEE Access, vol. 12, pp. 191586–191602, Dec. 2024, doi: 10.1109/ACCESS.2024.3516754.

5. D. J. Dsouza, A. P. Rodrigues and R. Fernandes, "Multi-Modal Comparative Analysis on Execution of Phishing Detection Using Artificial Intelligence," IEEE Access, vol. 12, pp. 163016–163028, Nov. 2024, doi: 10.1109/ACCESS.2024.3491429.

6. R. Banu and V. Vijayarani, "A Phishing Account Detection Model via Network Embedding for Ethereum," International Journal of Advanced Computer Science and Applications, vol. 15, no. 3, pp. 103–111, 2024.

7. J. Kim et al., "A Study on Adversarial Sample Resistance and Defense Mechanism for Multimodal Learning-Based Phishing Website Detection," IEEE Transactions on Information Forensics and Security, vol. 19, pp. 213–225, 2024.

8. R. C. Jain and S. Sharma, "DEPHIDES: Deep Learning Based Phishing Detection System," International Journal of Engineering Research & Technology (IJERT), vol. 13, no. 2, pp. 150–157, 2024.

9. K. V. Nithya and S. A. Meenakshi, "Phishing Webpage Detection: Unveiling the Threat Landscape and Investigating Detection Techniques," International Journal of Computer Applications, vol. 185, no. 4, pp. 30–37, 2024.

10. M. K. Kavitha, R. U. Rani and R. Rani, "An Efficient Detection of Phishing Website using Machine Learning," International Journal of Research Publication and Reviews, vol. 5, no. 1, pp. 19–25, 2024.