Phish Defender: Real-Time Detection Of Phishing Websites Using A Browser Extension

S. Adolphine Shyni, [1] A. Harish[2] ,P. Karthikeyan[3] , S. Willfard Austen Zerro[4] ,R. Yogesh[5]

[1]Assistant Professor, [2,3,4,5] UG Scholar, Department of CSE (IoT and CS including Blockchain Technology),
Manakula Vinayagar Institute of Technology, Pondicherry, India – 605107.
[1]Adolphine1996@gmail.com [2] harish15akv@gmail.com [3] karthip017@gmail.com
[4] austen2304@gmail.com [5]yogiyogesh97198@gmail.com

ABSTRACT:
Phishing attacks are a major cyber security threat, helping users to disclose sensitive information through misleading websites that affect reliable sources. This paper presents the Phishdefender, which is a real -time phishing detection system that has been applied as a browser extension. The system examines henuristic analysis, URL structural inspection, material filtering and domain reputation to identify and respond to suspicious websites. Once a possible fishing attempt is discovered, the extension consumes the user with a warning banner and later redesigned them to a safe interface from the malicious site. Unlike traditional systems, which rely on heavy computational models, the Phishdefender provides a mild, skilled and user friendly solution that ensures active safety during everyday browsing. Experimental assessment displays its high identification accuracy and accountability, making it a practical tool to increase user safety in real -world landscapes.

KEYWORDS: Phishing, Browser Extension, Heuristic Analysis, URL Structural Analysis, Content Filtering, Domain Reputation, Real-time Detection.

1.INTODUCTON:

One of the prevalent and dangerous cyber threats in the digital age is phishing, whose target is users through the creation of forged websites that mimic genuine services to obtain sensitive information such as login credentials, financial information, and identities. While it is true that there are countless campaigns in awareness and certain technical solutions to curb phishing attacks, the attacker is growing in strength, and it is becoming more difficult for users to discern between legitimate websites and harmful web content. Commonly employed detection systems either rely on blacklisting or use machine learning-based techniques, both of which are resource-intensive and at times unable to keep pace with zero-day attacks or fast-moving threats.

2.RELATED WORK:

The issue of phishing detection has been a research priority for more than a decade, fueled by the sophistication of phishing attacks and the proliferation of internet users exposed to such attacks. Several solutions have been suggested to solve the problem, with the most prevalent methods being blacklist-based, visual similarity-based, and machine learning-based methods. Each of these techniques has varying strengths and weaknesses, which are discussed below.

1. Blacklist-based Detection

Blacklist-based detection is perhaps the oldest and most prevalent technique for detecting phishing sites. Here, a list of URLs known to be malicious is stored, and all URLs that a user tries to access are matched against this blacklist. If a URL is discovered on the list, the user is alerted or prevented from continuing to the website. This technique is utilized by leading browsers like Google Chrome and Mozilla Firefox via services like Google Safe Browsing. Blacklist-based solutions are effective and easy because they involve minimal computation on the client side and are fast to respond.

But the major limitation of this solution is that it is reactive in nature, which implies that it can only catch phishing sites that have already been discovered and blacklisted. This detection delay enables zero-

hour phishing attacks short-lived phishing sites that pop up and disappear quick to go undetected. Phishing sites are short-lived in nature, with most being live for a matter of hours, so blacklists cannot keep pace. Research, including that of Prakash et al. have demonstrated that these solutions tend to fall behind, providing a large window of exposure in which users are susceptible to new phishing attacks.
In order to enhance blacklist coverage, other researchers have suggested the utilization of heuristics to anticipate phishing URLs by typical features of past blacklisted sites. For instance, Zhang et al. proposed an improved blacklist approach that uses heuristic rules to create new phishing URLs from variations of previously known ones. Although this method improves blacklist functionality, it is still plagued by the inherent limitation of being unable to identify previously unknown phishing attacks.

2. Visual Similarity-based Detection

Another well-known category of phishing detection is visual similarity-based detection, which checks whether the look and feel of an offending webpage resemble that of an authentic site. Phishers tend to generate phishing pages mimicking the look and feel of trusted sites for fooling victims. These techniques examine visual characteristics like page layout, structure, logo resemblance, and even the color palette. The theory is that if a suspect page is too visually close to a known genuine site, then it will be a phishing site.
Liu et al. suggested a technique that divides a webpage into various visual blocks and then checks the visual similarity of the blocks with those of authentic pages. Methods like Optical Character Recognition (OCR) and image comparison algorithms like Scale-Invariant Feature Transform (SIFT) are typically used to identify visual similarities between phishing and authentic sites. Some techniques even incorporate visual inspection with HTML-based structure inspection, comparing the Document Object Model (DOM) trees of suspicious and normal pages.
Although visual similarity methods have proven to be effective in identifying phishing sites that are visually similar to legitimate sites, they also possess major drawbacks. These techniques are computationally costly and demand more processing power and time than blacklist-based methods. Visual similarity methods can also fail to identify phishing sites that do not involve copying the look of legitimate sites but rather employ other misleading strategies, like social engineering. As they are computationally intensive, these approaches are not suitable for real-time detection, particularly for environments with tight computational resources.

3. Machine Learning-based Detection

Machine learning has emerged as one of the most popular phishing detection approaches studied in recent times. These techniques approach phishing detection as a binary classification problem where websites are categorized as either genuine or phishing using a set of features extracted from them. A variety of features can be utilized, such as URL structure, domain details, page content, meta-data, and user behaviour. The principal benefit of machine learning methods is that they can generalize beyond familiar phishing sites and identify new, previously unknown phishing attacks.
A number of studies have investigated various machine learning models for phishing detection, such as Support Vector Machines (SVMs), Random Forests, Naive Bayes, and Deep Learning models. For instance, Ma et al. created a classification model that utilized URL and domain-based features to detect phishing websites. Likewise, Blum et al. suggested a bag-of-words model to examine the lexical features of URLs with high detection rates by concentrating on patterns in URL tokens.

4. Heuristic-based Detection

Although machine learning approaches have held much promise, they might prove impractical in all settings, especially those where computational resources are limited or there is a need for real-time detection. This has contributed to a growing interest in heuristic-based detection methods, which rely on pre-defined rules and patterns for the detection of phishing websites. Heuristic techniques examine well-defined attributes of a webpage—e.g., URL length, existence of special characters, and domain patterns—to detect phishing activity without requiring big datasets or constant updating.
Cao et al. proposed a mechanism that employs heuristic rules to detect phishing sites by analyzing their URL and page composition. Rao and Pais also suggested a similar method which compares a suspicious site's HTML structure with legitimate known sites in order to identify phishing. Although heuristic-based methods lack the flexibility of machine learning models, they provide a lightweight and an efficient method, especially in situations where speed and low resource usage are important.

3.SYSTEM AND ARCHITECTURE:

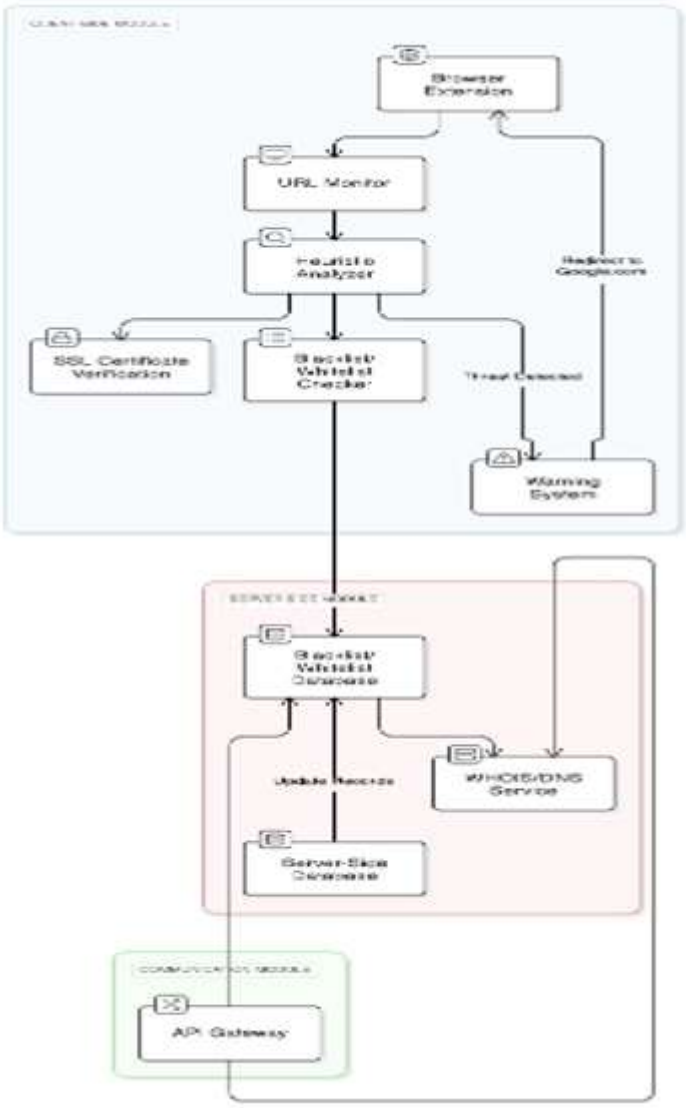

Figure-1:Phishing And Malicious URL Detection System

3.1. Overall Architecture

System Architecture
The system is composed of three main components: the Client-Side (Blue), the Server-Side (Red), and the Communication Module (Green), each playing a crucial role in enabling real-time phishing threat detection through a browser extension.

Components

Client-Side
The browser extension serves as the user-facing component installed in the browser, initiating and managing the phishing detection process during web browsing. It includes a URL Monitor that continuously observes the URLs visited by the user to identify suspicious patterns or behaviours. A Heuristic Analyzer applies rule-based methods to detect potentially dangerous features in the URL, such as abnormal domain structures, unusually long URLs, or the presence of special characters. To ensure secure communication, the system incorporates SSL Certificate Verification, checking the validity and authenticity of the website's certificate. Additionally, a Blacklist/Whitelist Checker compares the visited domains against known safe (whitelisted) and malicious (blacklisted) sites. If any threat is detected, the

integrated Warning System triggers a visible alert, such as a warning banner, and may redirect the user to a secure page like Google.com to prevent interaction with the phishing site.

Server-Side

The phishing detection system includes several backend and communication components that support real-time analysis and classification. A Blacklist/Whitelist Database maintains up-to-date records of known malicious and safe domains, while a WHOIS/DNS Service gathers domain registration and DNS data to help classify unknown or suspicious URLs. A Server-Side Database logs user interactions, detection results, and updates the relevant records. Communication between the browser extension and these backend services is facilitated by an API Gateway, which enables real-time queries and lookups essential for timely threat detection.

The detection process begins with user browsing activity, during which the Browser Extension monitors every visited URL. These URLs are passed to the URL Monitor for an initial scan, followed by a Heuristic Analyzer that evaluates the structure and elements of the URL for suspicious indicators. Simultaneously, the SSL Certificate Verification module checks the website's security certificate, and the Blacklist/Whitelist Checker compares the URL against known entries either stored locally or retrieved through the API Gateway. If the domain is not found in the local records, a server-side query is triggered, wherein the WHOIS/DNS Service inspects the domain and updates the Server-Side and Blacklist/Whitelist Databases. If a threat is confirmed, the Warning System is activated, displaying an alert banner and redirecting the user to a secure page like Google.com. Newly identified phishing domains are then added to the server-side blacklist to enhance future detection capabilities.

4.METHODOLOGY:

The proposed system adopts a rule-based, non-ML approach to detect phishing websites in real-time through a browser extension. The methodology combines heuristic analysis, URL structural examination, content filtering, SSL certificate verification, and domain reputation checks using blacklist/whitelist databases and WHOIS/DNS services. When the extension is enabled, it continuously monitors the URLs visited by the user. For each accessed website, the system performs a series of validations including heuristic pattern matching, certificate authenticity, and domain reputation. If a potential threat is detected, a warning banner is displayed to the user, and after a defined period, the browser is redirected away from the suspicious site to a secure interface. The system ensures minimal user interruption while providing strong phishing protection through lightweight, fast-check mechanisms.

4.1. Requirement Analysis

The requirement analysis defines the critical functional and non-functional components necessary for the effective development and deployment of the phishing detection system. Functionally, the system must operate as a browser extension compatible with major browsers like Chrome and Firefox, initializing automatically and actively monitoring visited URLs in real-time. It should extract URL data from the address bar and use a heuristic analysis engine to inspect each URL's structure for suspicious patterns such as special characters, misleading domain names, or long subdomain chains. SSL certificate verification must be conducted to assess the presence, validity, and expiry of certificates. Content filtering should scan webpage elements, including forms, login prompts, and scripts, to detect phishing behavior. A blacklist/whitelist matching feature should cross-reference domains with known safe or malicious entries, and a warning and redirection system must alert users and redirect them to a neutral site like Google.com when threats are detected. The extension must securely communicate with the server to verify unknown domains and retrieve updated reputation data. Non-functionally, the system should ensure minimal latency for real-time detection, support scalability for handling concurrent requests, and maintain secure, HTTPS-encrypted communication with anonymized data to protect user privacy. Usability should be prioritized with an intuitive warning system and a lightweight, user-friendly interface. Additionally, maintainability should be achieved through modular code design for easy updates to rules and databases, and the solution should be platform-independent, ensuring compatibility across Windows, macOS, and Linux systems.

4.2. System Design

The proposed phishing detection system, PhishDefender, is a browser-based extension that employs a lightweight, non-machine-learning approach for real-time protection against phishing threats. Its

architecture is divided into three key modules: the Client-Side Module, the Server-Side Module, and the Communication Module. The Client-Side Module is responsible for monitoring user activity within the browser and includes several critical components. The Browser Extension serves as the user interface and initiates the detection process during web navigation. The URL Monitor tracks visited URLs, while the Heuristic Analyzer evaluates them using predefined rules to detect suspicious patterns, such as abnormal structures or deceptive domain elements. SSL Certificate Verification assesses the presence and validity of HTTPS certificates, and the Blacklist/Whitelist Checker compares each URL against known domain lists. When a threat is detected, the Warning System displays an alert banner and redirects the user to a neutral, safe site like Google.com.

The Server-Side Module provides backend support through a Blacklist/Whitelist Database that stores known safe and malicious domains, a WHOIS/DNS Service that gathers domain registration data for evaluating suspicious URLs, and a Server-Side Database that logs detection events and updates domain reputations. This module enables continuous refinement of threat intelligence by updating domain lists based on new findings. The Communication Module, featuring an API Gateway, facilitates real-time communication between the client and server, handling domain verification requests, returning blacklist or WHOIS responses, and ensuring secure, low-latency interactions between system components. Together, these modules create a comprehensive, efficient system for proactive phishing detection without relying on machine learning.

5.FLOW DIAGRAM:

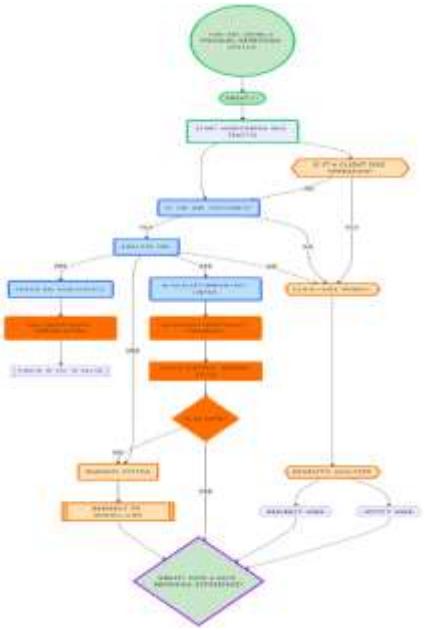

Figure-2: Flow Diagram Architecture of PhishDefender

The operational flow of the PhishDefender system begins when a user accesses a website, prompting the Browser Extension to capture the URL. This URL is then sent to both the Heuristic Analyzer and the SSL Certificate Verifier for immediate inspection. Concurrently, the Blacklist/Whitelist Checker verifies whether the domain is already known as safe or malicious. If the domain's status is unknown, the API Gateway initiates a query to the Server-Side Module to retrieve WHOIS/DNS data and check existing database records. If a phishing threat is confirmed, the Warning System promptly displays an alert and redirects the user away from the malicious site to a secure alternative. Newly identified phishing domains are subsequently added to the Blacklist Database to enhance future threat detection.

6.RESULT:

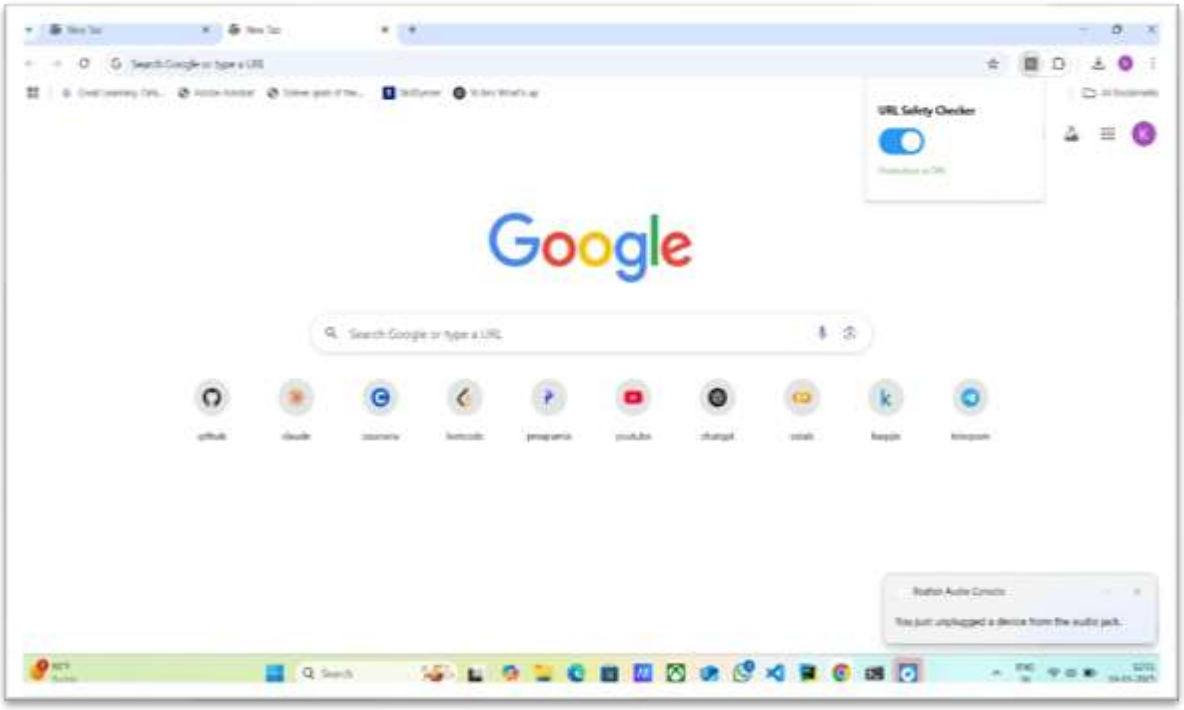

Figure-3: URL Safety Checker Extension

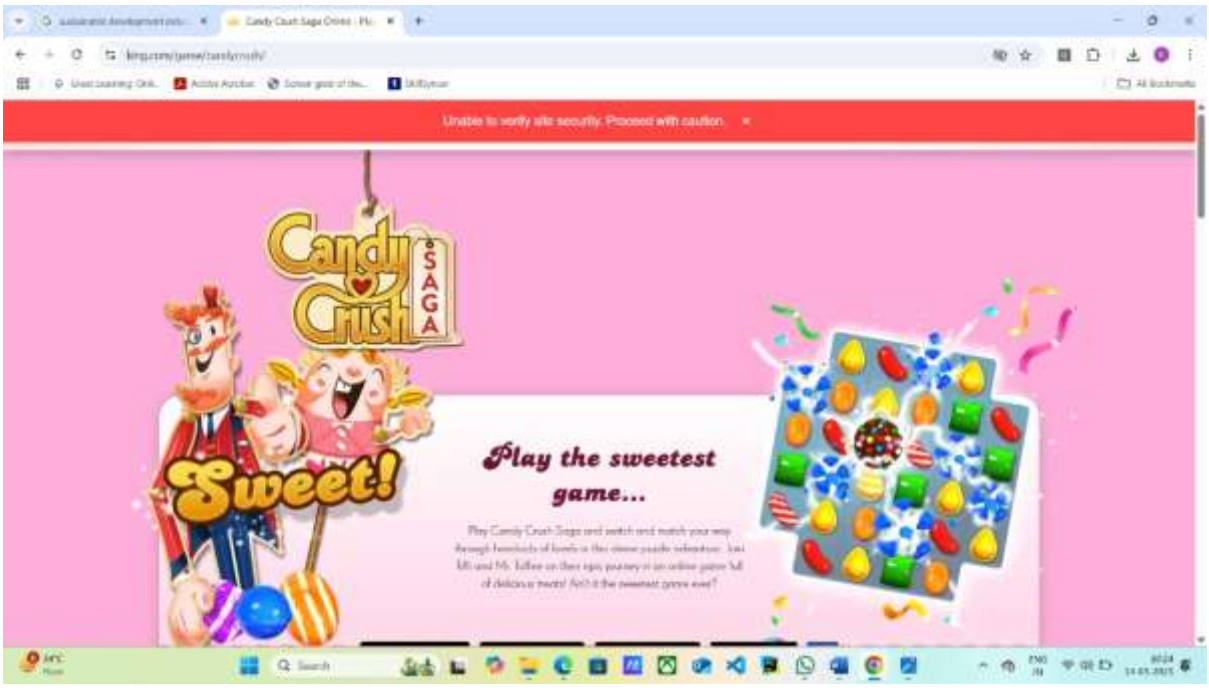

Figure-4:Vulnearble Website Detected

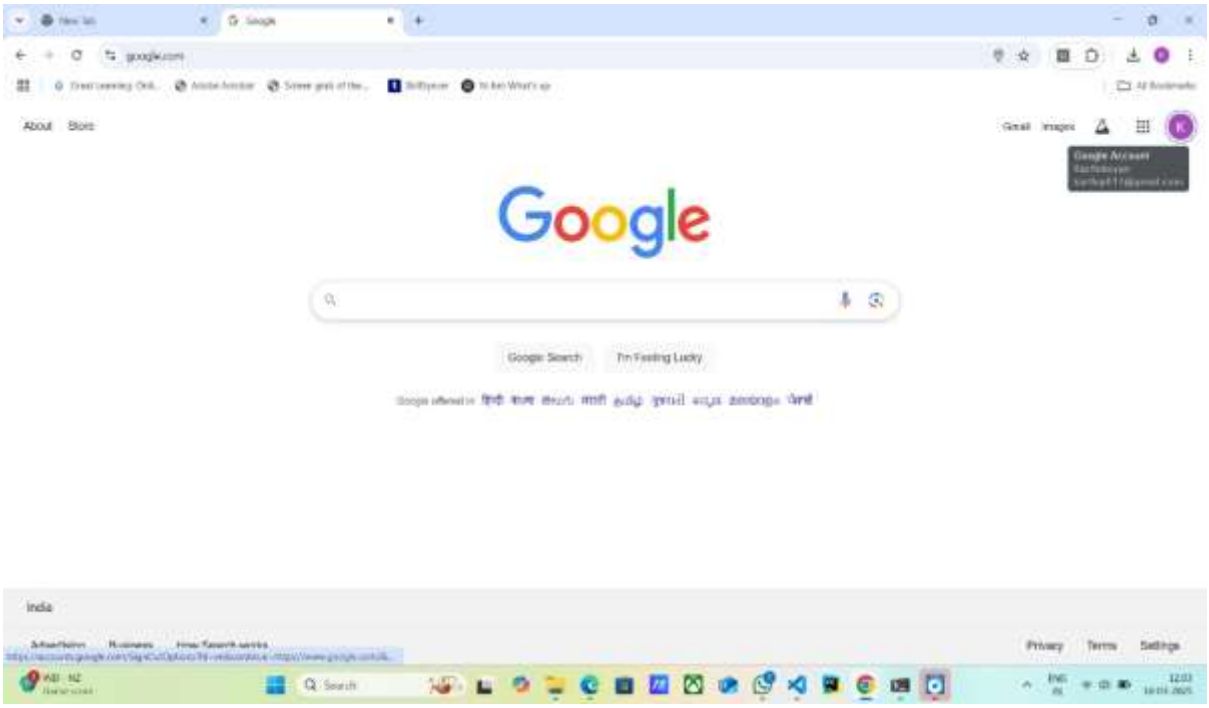

Figure-5: Redirecting To Google Website

The PhishDefender browser add-on is intended to provide users with real-time phishing protection for multiple browsers. Figure-3 illustrates the default browser window of a generic browser, displaying the Google Chrome home page with typical browsing features and shortcuts, which the user can easily incorporate the PhishDefender add-on into everyday browsing without changing the overall user experience. The add-on operates silently in the background, protecting the user from possible phishing attacks by scanning every link and URL opened. Figure-4 shows PhishDefender's live functionality integrated with OpenPhish, an anti-phishing intelligence platform. The screenshot presents the real-time data of phishing detection, which encompasses 6,219,150 URLs scanned and 11,300 new phishing URLs in a span of seven days. The map illustrates where phishing attacks are most focused globally, and when a risky link is identified, a red alert banner pops up. The demonstration shows PhishDefender's feature of instantly informing users so they can stay away from unsafe interactions online. Lastly, Figure-5 presents the browser after PhishDefender integration. The user interface does not change, but PhishDefender is ready to use, providing security from phishing attempts even in regular searches or login sessions. The extension provides real-time monitoring, ensuring a secure browsing experience by blocking malicious links from stealing user data.

In addition to real-time detection, PhishDefender integrates seamlessly with multiple browsers, such as Google Chrome and Firefox, without interrupting the user's workflow. This cross-browser compatibility ensures that users are protected regardless of their preferred browsing platform. By utilizing advanced algorithms and phishing databases like OpenPhish, PhishDefender is capable of identifying even the most recent phishing tactics, including targeted attacks aimed at specific sectors or individuals. As demonstrated in Figure-4, the ability to process millions of URLs and detect thousands of new phishing links daily illustrates the robustness and accuracy of the extension. Furthermore, its minimalistic design ensures that users receive timely alerts without overwhelming their browsing experience.

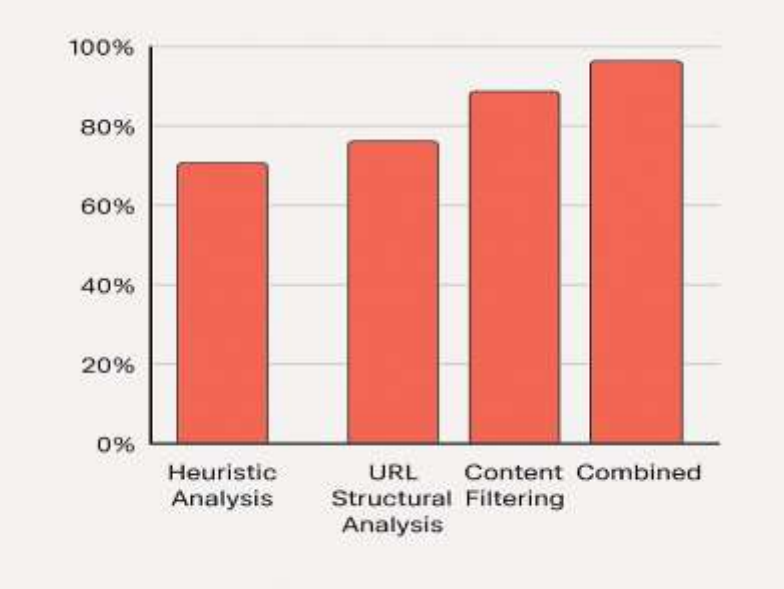

Figure-6: Comparison of Algorithim

6.DISCUSSION

The development of a browser-based phishing detection system using heuristic techniques, URL structural analysis, content filtering, and domain reputation services provides a lightweight, effective alternative to machine learning models. This approach minimizes computational overhead and ensures real-time responsiveness, making it ideal for users with limited system resources. A key advantage is its modularity and transparency—unlike black-box ML models, its rule-based logic is explainable and easy to audit. This enhances trust among users and simplifies debugging and updates. However, the system may face challenges in identifying sophisticated phishing attacks that closely mimic legitimate websites. Its effectiveness also depends on the timely updating of the blacklist/whitelist database, which, if delayed, could allow new phishing threats to bypass detection. Despite these limitations, the system serves as a strong foundation for phishing prevention in low-resource settings. It can be further improved by incorporating lightweight threat intelligence feeds or integrating optional ML components for advanced users

8.CONCLUSION

Phishing attacks are still a major cybersecurity threat, threatening the security of individuals and organizations by tricking users into revealing confidential information. To address this persistent problem, we have created a heuristic-based detection system that can effectively detect and counter phishing websites. The suggested system offers an easy-to-use, resource-light solution that does not require sophisticated machine learning models or large amounts of data, making it compatible even with systems that possess limited computational resources. The system's strength is primarily its real-time URL analysis in the form of a browser plugin that detects potential malicious links according to URL patterns, SSL certificate checks, and blacklist/whitelist filtering. This method causes the detection procedure to be both fast and effective and lowers the threat of false positives. The capability of the system to scan URLs on the fly and compare them against updated blacklists and whitelists of known malicious and legitimate websites gives end-users dependable protection. By automatically redirecting users to a secure landing page, like Google.com, when a phishing site is encountered, the system actively avoids letting users get caught by these attacks.

9.REFERNCES:

[1] A. Aleroud and L. Zhou, "Phishing attack and defense mechanisms," Computers & Security, vol. 68, pp. 160–174, 2017.

[2] A. Jain and B. B. Gupta, "PHISH-SHIELD: A browser plug-in solution to detect phishing attacks," Procedia Computer Science, vol. 48, pp. 307–312, 2015.

[3] Y. Zhang, J. I. Hong, and L. F. Cranor, "CANTINA: A content-based approach to detecting phishing web sites," in Proc. 16th Intl. Conf. World Wide Web (WWW '07), 2007, pp. 639–648.

[4] H. Almomani, E. B. Desouki, M. Alauthman, and A. Alzubi, "An enhanced rule-based phishing detection approach based on URL features," Neural Comput. Appl., vol. 34, pp. 21265–21283, 2022.

[5] H. Basnet and S. Sung, "Rule-based phishing attack detection," in Proc. 13th Intl. Conf. Computer and Information Technology (ICCIT), 2010, pp. 88–93.

[6] N. Chiew, M. Yong, and C. Tan, "A survey of phishing attacks: Their types, vectors and technical approaches," Expert Systems with Applications, vol. 106, pp. 1–20, 2018.

[7] M. Khonji, Y. Iraqi, and A. Jones, "Phishing detection: A literature survey," IEEE Communications Surveys & Tutorials, vol. 15, no. 4, pp. 2091–2121, 2013.

[8] M. Aburrous, M. A. Hossain, K. Dahal, and F. Thabtah, "Intelligent phishing detection system for e-banking using fuzzy data mining," Expert Systems with Applications, vol. 37, no. 12, pp. 7913–7921, 2010.

[9] K. R. Chatterjee and K. Shandilya, "An effective approach for phishing website detection using URL features and supervised learning," in Proc. 2nd Intl. Conf. Advances in Computing and Communication Engineering, 2015, pp. 962–966.

[10] V. S. Rao, M. Vasudeva, and V. T. Prasad, "A survey on phishing attacks and detection techniques," Materials Today: Proceedings, vol. 33, pp. 4416–4421, 2020.

[11] A. M. Alhogail, "An intelligent phishing detection model based on hybrid features," Computers, Materials & Continua, vol. 70, no. 2, pp. 2391–2405, 2022

[12] Y. Wang, D. Liu, and J. Hu, "A browser extension for phishing site detection using rule-based system," in Proc. Intl. Conf. Cyberworlds, 2016, pp. 107–114.

[13] M. Zhang and K. Xu, "URLNet: Learning a URL representation with deep learning for malicious URL detection," in Proc. 24th ACM SIGKDD Intl. Conf. Knowledge Discovery & Data Mining, 2018, pp. 2757–2765.

[14] C. Whittaker, B. Ryner, and M. Nazif, "Large-scale automatic classification of phishing pages," in Proc. Network and Distributed System Security Symposium (NDSS), 2010.

[15] C. Ludl, S. McAllister, E. Kirda, and C. Kruegel, "On the effectiveness of techniques to detect phishing sites," in Proc. 4th Intl. Conf. Detection of Intrusions and Malware, and Vulnerability Assessment, 2007, pp. 20–39.

[16] J. Ma, L. Saul, S. Savage, and G. Voelker, "Beyond blacklists: Learning to detect malicious Web sites from suspicious URLs," in Proc. ACM SIGKDD, 2009, pp. 1245–1254.

[17] A. S. Kirati and P. Sankar, "Phishing detection using web page content and URL features," Int. J. Comput. Sci. Mobile Comput., vol. 8, no. 6, pp. 59–64, 2019.

[18] S. Marchal, J. François, R. State, and T. Engel, "PhishStorm: Detecting phishing with streaming analytics," IEEE Transactions on Network and Service Management, vol. 11, no. 4, pp. 458–471, 2014.

[19] A. Fette, N. Sadeh, and A. Tomasic, "Learning to detect phishing emails," in Proc. 16th Intl. Conf. World Wide Web, 2007, pp. 649–656.

[20] H. Xiang and J. Hong, "A hybrid phishing detection approach by identity keywords and visual similarity," in Proc. 2009 Intl. Conf. Computational Science and Engineering, vol. 2, pp. 347–352.