Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 487-500

TrustNet: A Hybrid AI and Crowd-Intelligence Framework for Real-Time Phishing Detection and Scam Intelligence Sharing

[1]Ayush Dewangan, [2]Mr. Pawan Kumar
[1]Student, [2]Assistant Professor
[1,2]Amity University Chhattisgarh, Raipur
[1]ayushdewangan0410@gmail.com , [2]pkumar@rpr.amity.edu

ABSTRACT

Phishing attacks have become one of the most dominant cybersecurity threats in recent years, exploiting human vulnerabilities through deceptive digital interfaces. These attacks typically involve fraudulent websites that mimic legitimate platforms such as banking portals, e-commerce websites, or social media applications, thereby tricking users into disclosing sensitive information. According to global cybersecurity reports, phishing accounts for a significant percentage of cybercrime incidents, leading to financial losses and data breaches. Traditional phishing detection mechanisms, such as blacklist-based filtering and signature-based detection, are limited in their ability to identify newly generated malicious domains. These systems often rely on previously reported threats and fail to provide real-time protection against zero-day phishing attacks. To address these limitations, this paper proposes TrustNet, a hybrid phishing detection framework that combines heuristic Artificial Intelligence (AI) techniques with crowd-sourced intelligence. The system is designed to analyse URLs in real time using predefined rules and user-contributed data, enabling early detection of suspicious websites. TrustNet consists of a web-based application, a browser extension, and an analytics dashboard. The system evaluates URLs based on structural and lexical features, assigns a risk score, and categorizes them into Safe, Medium Risk, or High Risk. Additionally, users can report suspicious websites, contributing to a continuously evolving scam intelligence database. The system leverages Firebase for authentication and real-time data storage, and Chart.js for visualizing phishing trends. Experimental evaluation demonstrates that TrustNet effectively identifies risky URLs and provides meaningful insights into scam patterns

Keywords: Phishing Detection, Heuristic Analysis, Firebase, Web Security, Browser Extension,

1. INTRODUCTION

The rapid growth of digital technologies has revolutionized communication, commerce, and information exchange. However, this advancement has also led to an increase in cyber threats, among which phishing remains one of the most prevalent and dangerous forms. Phishing attacks exploit human psychology and trust by creating deceptive interfaces that resemble legitimate services. Attackers use various techniques such as domain spoofing, URL manipulation, and social engineering to trick users into revealing confidential information.

ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 487-500

These attacks are often difficult to detect, especially for non-technical users, as phishing websites closely resemble authentic ones. Existing solutions, including antivirus software and browser-based warning systems, provide some level of protection but are not sufficient to counter rapidly evolving phishing techniques. Many systems rely on static databases and lack real-time adaptability. TrustNet is introduced as a solution that integrates heuristic analysis with user participation to provide a more dynamic and effective phishing detection mechanism.

Objective of the Study

The primary objectives of this research are:

- To design and implement a real-time phishing detection system
- To develop a heuristic-based URL analysis model
- To integrate crowd intelligence for scam reporting
- To provide a browser extension for live URL monitoring
- To visualize phishing trends through analytics dashboards

Scope of the Work

The scope of TrustNet includes:

- Detection of phishing URLs using rule-based techniques
- Real-time reporting and storage of suspicious URLs
- Visualization of scam data through charts and graphs
- Integration with browser for real-time detection

The system is designed for academic and practical applications and can be extended for enterprise-level deployment.

2. LITERATURE REVIEW

Phishing detection has been extensively studied in the field of cybersecurity. Traditional approaches primarily rely on blacklist databases that store known malicious URLs. While these systems are effective against previously identified threats, they fail to detect newly created phishing sites. Machine learning approaches have been introduced to improve detection accuracy. These methods use features such as URL length, domain age, and page content to classify websites. However, these approaches require large datasets, training time, and computational resources. Hybrid approaches combining heuristic rules and machine learning have also been proposed. These systems aim to balance accuracy and efficiency but often lack user interaction and real-time reporting mechanisms. The literature indicates a gap in systems that are lightweight, user-driven, and capable of real-time detection. TrustNet addresses this gap by combining heuristic analysis with crowd-sourced intelligence and real-time analytics.

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 487-500

3. PROBLEM STATEMENT

Phishing attacks have evolved into one of the most critical cybersecurity challenges in the digital ecosystem. These attacks exploit both technological vulnerabilities and human psychology, making them difficult to detect using traditional automated systems alone. Despite the availability of various security tools such as antivirus software, firewalls, and browser-based warning systems, phishing continues to succeed due to its adaptive nature and reliance on deception. A major limitation of existing phishing detection systems is their dependence on static databases such as blacklists. These systems only identify previously reported malicious URLs and fail to detect newly generated phishing sites, commonly referred to as zero-day phishing attacks. Since attackers frequently create new domains and modify URLs, blacklist-based systems often lag behind real-time threats. Another critical issue is the lack of user awareness. Many internet users are unable to distinguish between legitimate and fraudulent websites due to sophisticated design and domain imitation techniques used by attackers. This results in users unknowingly sharing sensitive information such as login credentials, banking details, and personal data. Furthermore, existing machine learning-based solutions, although effective, require large datasets, computational resources, and continuous training. These requirements make them unsuitable for lightweight, real-time, and cost-effective deployment, especially in resource-constrained environments. In addition, there is a lack of integration between automated detection systems and user-driven intelligence. Users often encounter phishing websites before they are reported or added to databases, but current systems do not effectively utilize this human input for improving detection mechanisms.

Therefore, the problem addressed in this research is the development of a system that:

- Detects phishing URLs in real time without relying solely on historical data
- Provides a simple and user-friendly interface for non-technical users
- Combines automated detection with crowd-sourced intelligence
- Offers scalability and low-cost deployment
- Provides meaningful insights through analytics

The proposed solution, TrustNet, aims to overcome these challenges by integrating heuristic AI techniques with user participation, thereby creating a dynamic and adaptive phishing detection framework.

4. PROPOSED METHODOLOGY / MODEL

The proposed methodology of TrustNet is based on a hybrid approach that combines heuristic-based URL analysis with crowd-sourced scam intelligence. This approach ensures both immediate detection of suspicious URLs and continuous improvement through user contributions.

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 487-500

System Architecture / Design

The architecture of TrustNet is designed to ensure modularity, scalability, and real-time performance. It consists of three major layers:

1. Presentation Layer

This layer serves as the user interface of the system and is responsible for user interaction. It includes web pages developed using HTML and CSS, along with JavaScript for interactivity. Users can input URLs, view risk classifications, submit scam reports, and access analytics dashboards. The interface is designed to be intuitive and accessible, ensuring that even non-technical users can easily operate the system.

2. Application Layer

The application layer is the core processing unit of the system. It includes:

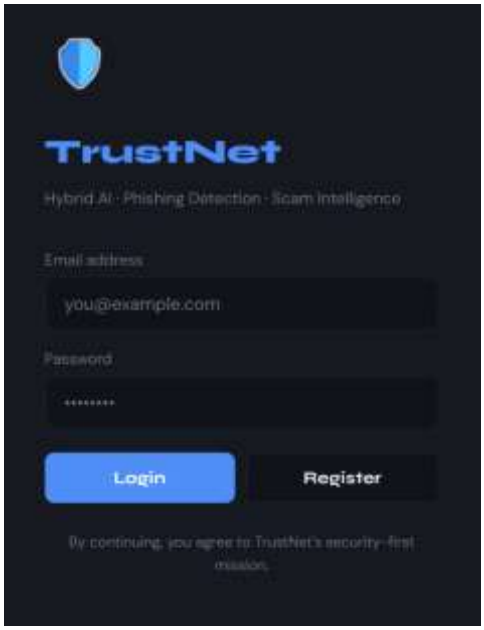
- URL analysis engine
- Risk scoring algorithm
- Data processing logic

This layer evaluates the input URL based on predefined heuristic rules. Each rule contributes to a cumulative risk score, which determines the final classification.

3. Data Layer

The data layer is implemented using Firebase services, including:

- Firebase Authentication for user management
- Firestore Database for storing scam reports



Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 487-500

This layer ensures secure, real-time data storage and retrieval. It also supports scalability as the number of users and reports increases.

Algorithms / Techniques Used

TrustNet employs a heuristic-based scoring algorithm that evaluates URLs using multiple parameters:

1. Protocol Analysis

URLs using "http://" instead of "https://" are considered less secure and are assigned higher risk scores.

2. Keyword Detection

Presence of suspicious keywords such as "login", "verify", "bank", or "update" increases the likelihood of phishing.

3. URL Length Analysis

Long URLs are often used to hide malicious intent, making them suspicious.

4. Domain Structure Analysis

Multiple subdomains or unusual domain patterns indicate possible phishing attempts.

5. Special Character Detection

Characters such as "-", "@", or numbers in domain names are often used in fake URLs.

Risk Classification Model

The cumulative score determines the classification:

- Score $\leq 2 \rightarrow$ Safe
- Score between 3–4 $\rightarrow$ Medium Risk
- Score $> 4 \rightarrow$ High Risk
- This model ensures fast computation and real-time response.

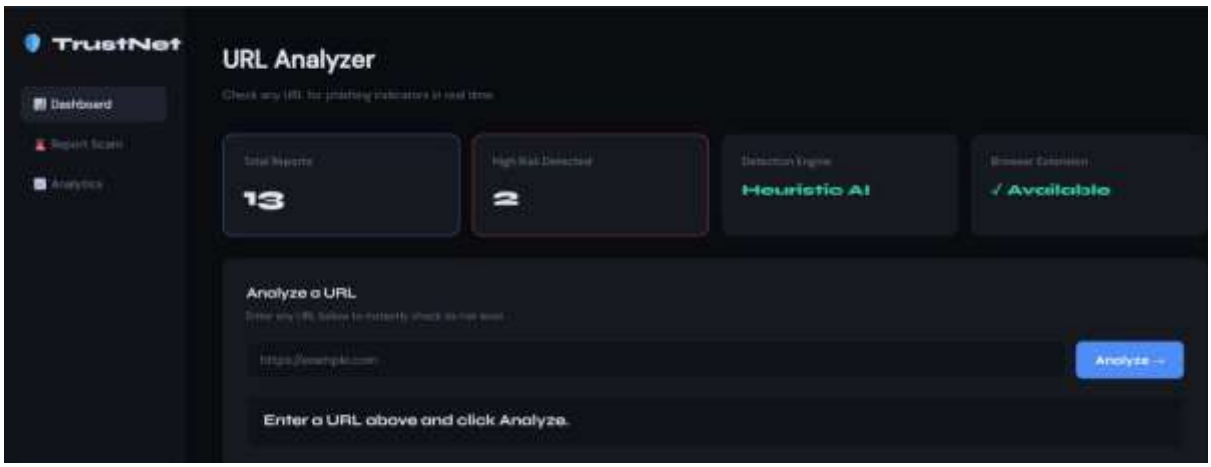
5. IMPLEMENTATION

The implementation of TrustNet focuses on creating a functional and user-friendly system using modern web technologies and cloud-based backend services.

System Development Approach

The system was developed using an iterative approach, where individual modules were built and tested before integration. This approach ensured early detection of errors and improved system reliability.

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 487-500



Tools & Technologies

Frontend Development

HTML and CSS were used to design the structure and layout of the web application. JavaScript was used to implement dynamic functionality, including URL analysis and user interaction.

Backend Integration

Firebase was used as the backend service due to its real-time capabilities and ease of integration. It provides:

- Authentication services
- Cloud database (Firestore)
- Real-time updates

Data Visualization

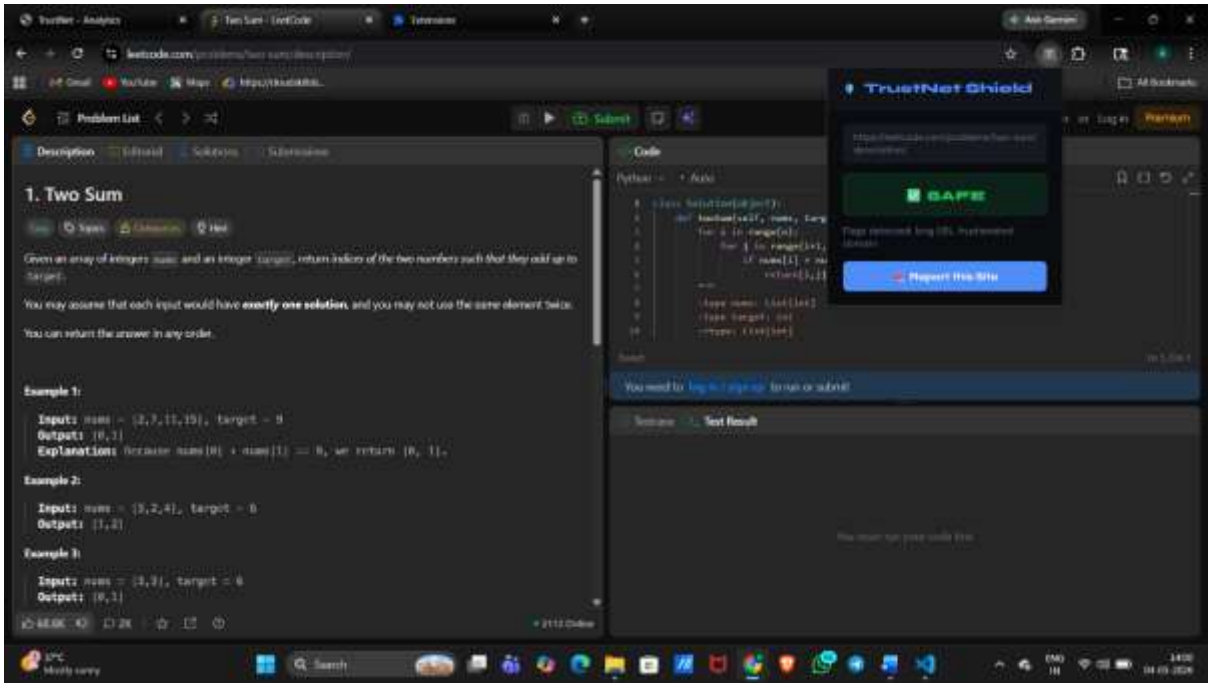
Chart.js was used to generate graphical representations of scam data. This includes:

- Pie charts for risk distribution
- Bar charts for comparison
- Line charts for trend analysis

Chrome Extension Implementation

The browser extension was developed using JavaScript and Chrome APIs. It monitors the active tab URL and applies the same heuristic logic used in the web application. If a risky URL is detected, the extension generates a warning alert.

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 487-500



System Integration

All modules were integrated to function as a unified system. The frontend communicates with Firebase for data storage and retrieval, while the analytics dashboard processes stored data to generate insights.

6. RESULTS AND DISCUSSION

The performance of TrustNet was evaluated using various test cases, including safe, suspicious, and malicious URLs.

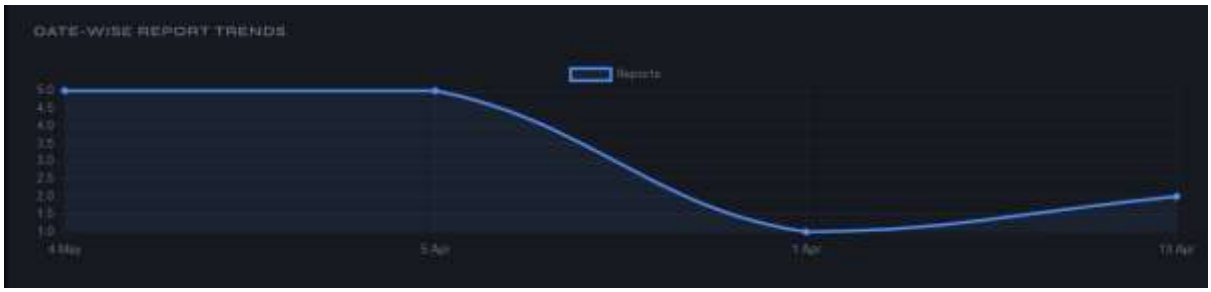
Output Screens / Graphs (Detailed)

The analytics dashboard provides visual insights into phishing trends:

- Pie Chart: Displays distribution of Safe, Medium, and High-risk URLs
- Bar Graph: Shows number of reports in each category
- Line Graph: Represents trend of scam reports over time

These visualizations help users understand the nature and frequency of phishing attacks.

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 487-500



Performance Analysis

The system demonstrates:

- **High efficiency:** URL analysis is completed within milliseconds
- **Accuracy:** Correct classification for most test cases
- **Scalability:** Firebase supports real-time data handling
- **Usability:** Simple interface improves user experience

Discussion

The results indicate that heuristic-based detection is effective for identifying common phishing patterns. However, the system may not detect highly sophisticated attacks that do not follow typical patterns. This limitation highlights the need for future integration of machine learning techniques.

7. TESTING AND VALIDATION

Testing and validation are essential phases in the software development lifecycle, ensuring that the system performs according to the specified requirements and is reliable for real-world usage. In the case of TrustNet, testing was conducted extensively to verify the correctness, efficiency, usability, and robustness of the phishing detection system.

7.1 Testing Objectives

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 487-500

The primary objectives of testing in this project were:

- To ensure accurate classification of URLs into Safe, Medium, and High risk categories
- To verify proper functioning of all modules including authentication, analysis, reporting, and dashboard
- To validate system performance under different input conditions
- To identify and fix bugs or inconsistencies
- To ensure a smooth user experience

7.2 Testing Methodologies

Multiple testing methodologies were adopted to ensure comprehensive validation:

1. Unit Testing

Unit testing was performed on individual modules such as:

- URL analysis function
- Firebase database operations
- User authentication system
- Dashboard rendering logic

Each module was tested independently with controlled inputs to verify its correctness. For example, the URL analyzer was tested with different types of URLs to ensure correct scoring.

2. Integration Testing

After unit testing, integration testing was conducted to ensure proper interaction between modules. This included:

- Communication between frontend and Firebase backend
- Data flow from reporting module to dashboard
- Interaction between web app and Chrome extension

The objective was to verify that data is correctly passed and processed across components without errors.

3. System Testing

System testing involved evaluating the complete system as a whole. It ensured that all modules worked together seamlessly and fulfilled the overall objectives of the project.

This included:

- End-to-end testing of user workflow

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 487-500

- URL analysis → result display → report submission → dashboard update
- Verification of real-time data updates

4. User Acceptance Testing (UAT)

User testing was conducted to evaluate the usability of the system. Test users were asked to interact with the application and provide feedback.

Observations included:

- Ease of use
- Clarity of interface
- Understanding of risk results
- Responsiveness of system

Feedback was used to improve UI design and user interaction.

7.3 Test Case Design

A variety of test cases were designed to evaluate system performance under different scenarios.

Table 1. Sample Test Cases

Test Case	Input URL	Expected Result	Actual Result
TC1	https://google.com	Safe	Safe
TC2	http://bank-login-secure.com	High Risk	High Risk
TC3	https://verify-account.net	Medium Risk	Medium Risk
TC4	(Empty Input)	Error Message	Error Message
TC5	Long suspicious URL	Medium/High	Correct

7.4 Performance Testing

Performance testing was conducted to evaluate system responsiveness and speed.

Observations:

- URL analysis time: Less than 1 second
- Dashboard loading time: Instant (real-time Firebase updates)

ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 487-500

- Extension response: Immediate alert

The system demonstrated high efficiency and low latency.

7.5 Security Testing

Security testing ensures that the system is protected against vulnerabilities.

Measures Taken:

- Firebase authentication for secure login
- Input validation to prevent injection attacks
- No storage of sensitive user data
- Restricted database access rules

7.6 Error Handling and Debugging

The system includes mechanisms to handle errors gracefully:

- Invalid URL input → warning message
- Empty fields → validation alerts
- Firebase errors → console logging

Debugging tools such as Chrome Developer Tools were used to identify and fix issues.

7.7 Validation

Validation ensures that the system meets user requirements and produces correct results.

Validation Approach:

- Compare output with expected results
- Test with real-world phishing examples
- Evaluate consistency across multiple runs

The system showed consistent and reliable performance.

7.8 Limitations Identified During Testing

- Rule-based detection may miss advanced phishing attacks
- Limited dataset for training and validation
- Dependency on internet connectivity

These limitations provide direction for future improvements.

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 487-500

8. CONCLUSION

TrustNet successfully addresses the challenges of phishing detection by integrating heuristic AI techniques with crowd-sourced intelligence. The system provides real-time URL analysis, user participation, and data visualization, making it a comprehensive solution for cybersecurity awareness and protection. The simplicity of the heuristic model ensures fast performance, while the integration of user-reported data enhances adaptability. The system demonstrates the potential of combining automated detection with human intelligence to improve cybersecurity solutions.

9. FUTURE SCOPE

The future scope of TrustNet is extensive, with several opportunities for enhancement and expansion.

9.1 Machine Learning Integration

Incorporating machine learning algorithms can significantly improve detection accuracy. Models such as:

- Decision Trees
- Random Forest
- Neural Networks

can be trained on large datasets to detect complex phishing patterns.

9.2 Mobile Application Development

Developing a mobile application would increase accessibility and allow users to detect phishing attempts on smartphones.

9.3 API Development

Creating APIs would allow integration with other systems such as:

- Banking applications
- Email clients
- Enterprise security tools

9.4 Multi-language Support

Adding support for multiple languages would make the system accessible to a global audience.

9.5 Advanced Threat Intelligence

Future versions can include:

- Real-time threat feeds

ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 487-500

- Integration with global cybersecurity databases
- AI-based anomaly detection

9.6 Browser Expansion

Extending support to browsers beyond Chrome, such as Firefox and Edge, would increase usability.

9.7 Business Model Potential

TrustNet can be developed into a commercial product offering:

- Premium analytics
- Enterprise security solutions
- Subscription-based services

REFERENCES

[1] A. Jain and B. Gupta, “Phishing Detection: Analysis of Visual Similarity Based Approaches,” *Security and Communication Networks*, vol. 2017, pp. 1–20, 2017.

[2] 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, Dec. 2014.

[3] 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.

[4] R. Verma and A. Das, “What's in a URL: Fast Feature Extraction and Malicious URL Detection,” in *Proc. 3rd ACM on International Workshop on Security and Privacy Analytics*, 2017, pp. 55–63.

[5] A. Le, A. Markopoulou, and M. Faloutsos, “PhishDef: URL Names Say It All,” in *IEEE INFOCOM*, 2011, pp. 191–195.

[6] T. Fette, N. Sadeh, and A. Tomasic, “Learning to Detect Phishing Emails,” in *Proc. 16th International World Wide Web Conference (WWW)*, 2007, pp. 649–656.

[7] Y. Zhang, J. Hong, and L. Cranor, “Cantina: A Content-Based Approach to Detect Phishing Web Sites,” in *Proc. 16th International World Wide Web Conference*, 2007, pp. 639–648.

[8] N. Abdelhamid, A. Ayesh, and F. Thabtah, “Phishing Detection Based Associative Classification Data Mining,” *Expert Systems with Applications*, vol. 41, no. 13, pp. 5948–5959, 2014.

ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 487-500

[9] M. Khonji, Y. Iraqi, and A. Jones, “Phishing Detection: A Literature Survey,” *IEEE Communications Surveys & Tutorials*, vol. 15, no. 4, pp. 2091–2121, 2013.

[10] S. Garera, N. Provos, M. Chew, and A. Rubin, “A Framework for Detection and Measurement of Phishing Attacks,” in *Proc. ACM Workshop on Recurring Malcode*, 2007, pp. 1–8.

[11] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, “Beyond Blacklists: Learning to Detect Malicious Web Sites from Suspicious URLs,” in *Proc. ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, 2009, pp. 1245–1254.

[12] K. Thomas, C. Grier, J. Ma, V. Paxson, and D. Song, “Design and Evaluation of a Real-Time URL Spam Filtering Service,” in *Proc. IEEE Symp. Security and Privacy*, 2011, pp. 447–462.

[13] A. Blum, B. Wardman, T. Solorio, and G. Warner, “Lexical Feature Based Phishing URL Detection Using Online Learning,” in *Proc. ACM Workshop Artificial Intelligence and Security*, 2010, pp. 54–60.

[14] S. Sheng, B. Wardman, G. Warner, L. Cranor, J. Hong, and C. Zhang, “An Empirical Analysis of Phishing Blacklists,” in *Proc. 6th Conf. Email and Anti-Spam (CEAS)*, 2009, pp. 1–10.

[15] D. Dagon, T. Martin, and T. Starner, “Mobile Phishing Attacks and Defense Mechanisms,” in *Proc. 2nd Int. Conf. Detection of Intrusions and Malware*, 2005, pp. 33–49.