IJIRIS: International Journal of Innovative Research in Information Security
Volume 12, Issue 04, April 2026
E-ISSN: 2349-7017
P-ISSN: 2349-7009
https://www.ijiris.com/archives
https://doi.org/10.26562/ijiris.2026.v1204.06

PhishDetect Pro: A Servlet Based Smart Wallet Simulation and Approval Phishing Detection Frame work Using Intent Validation

B.Ratnamala
Assistant Professor, Department of CSE
Guru Nanak Institute of Technology, Hyderabad, Telangana, India
https://orcid.org/0009-0004-5360-7540
Tadam Sravan, Pathi Harsha Vardhan Reddy, Rohith Rejinthla
Students, Department of CSE
Guru Nanak Institute of Technology, Hyderabad, Telangana, India

Publication History
Manuscript Reference No: IJIRIS/RS/Vol.12/Issue04/ISAP26.ISAP10085
Research Article | Open Access | Double-Blind Peer-Reviewed| ArticleID: IJIRAE/RS/Vol.12/Issue04/ISAP26.ISAP10085
Received:02,March 2026, Revised: 29, March 2026, Accepted: 10,April 2026,Published Online: 22, April 2026.
https://www.ijiris.com/volumes/Vol12/iss-04/06.ISAP26.ISAP10085.pdf
Article Citation:Ratnamala,Tadam,Pathi,Rohith(2026),PhishDetect Pro: A Servlet Based Smart Wallet Simulation and Approval Phishing Detection Frame work Using Intent Validation. IJIRIS: International Journal of Innovative Research in Information Security, Volume 12, Issue 04 of 2026 pages 296-302
Doi:> https://doi.org/10.26562/ijiris.2026.v1204.06 **BibTeX Key: Ratnamala@2026**
IJIRIS papers should be cited as IJIRIS(International Journal of Innovative Research in Information Security, AM Publications, India 2026, ISSN 2349-7017, https://doi.org/10.26562/ijiris.2026.v1204.06 The journal's official abbreviation is IJIRIS. Orcid: https://orcid.org/0009-0004-9398-7488

Abstract: This project investigates a highly deceptive blockchain scam known as approval phishing, where users are tricked into unknowingly granting access to their wallet funds. The fraud is executed via a malicious smart contract architecture (SCA) in which the user unknowingly approves a contract controlled by an attacker. This contract is then triggered by a malicious external owned account (EOA), allowing the unauthorized transfer of tokens. The attack is exacerbated by wallet UI weaknesses, which often fail to adequately present risks associated with approval requests. The scam infrastructure includes a comprehensive management system hosted on a secured web server, often hidden behind CDN layers to avoid blocklisting. This system not only handles the fake investment interface shown to victims but also manages scammer operations including statistics, fund withdrawals, and dynamic content manipulation. This project aims to simulate and analyze these attacks through a Servlet-JSP-based web application, evaluate wallet vulnerabilities, detect scam contract behavior, and propose smart wallet enhancements inspired by EIP-4337-based account abstraction. A comparative study and approval analysis system is also implemented to educate users and recommend
Keywords: Approval Phishing, Smart Wallet Simulation, Intent Validation, Risk Detection (SIVRD), Token Drain Simulation, Blockchain Ledger, Smart Contracts, Servlet-Based Web Application, Phishing Detection, Transaction Security.

I. INTRODUCTION

PhishDetect Pro is a project designed to study, detect, and prevent approval phishing attacks in blockchain-based wallet systems. In today's digital world, blockchain technology and cryptocurrency are becoming very popular for online transactions and investments. Many people use digital wallets to store, send, and receive tokens. But as the number of users increases, online scammers are also finding new ways to cheat people. One of the most common and dangerous scams is called approval phishing. Approval phishing happens when a hacker tricks a user into approving a fake smart contract. This fake contract secretly gives the hacker full control over the user's wallet. Once the user clicks on the approve button, the attacker can later transfer all the tokens without any further permission. The main reason this happens is because most wallet applications do not clearly show what the user is approving. Users think they are approving a normal transaction, but in reality, they are allowing someone else to control their funds.

II. LITERATURE SURVEY

J. Liu, J. Chen, J. Wu, Z. Wu, J. Fang, and Z. Zheng(2024)

This paper focuses on uncovering Ethereum-based phishing gangs by deeply analysing blockchain data, transaction flows, and malicious approval patterns. It investigates how organized fraud groups execute approval phishing attacks, where victims unknowingly authorize attackers to transfer their tokens. By studying both on-chain and off-chain operations, the project aims to understand how phishing websites, fake investment platforms, and malicious smart contracts function together as a coordinated scam ecosystem. Transaction clustering techniques, address linkage, and behaviour modelling are used to identify attacker-controlled wallets and their fund movement strategies. The project also simulates real approval-based scams using a Servlet–JSP web application to demonstrate how victims fall prey to deceptive UI flows. Detailed monitoring is conducted on attacker contracts, phishing links, and CDN-protected scam servers to map the entire fraud infrastructure

E-ISSN: 2349-7017
P-ISSN: 2349-7009
https://www.ijiris.com/archives
https://doi.org/10.26562/ijiris.2026.v1204.06

R. Liang, J. Chen, K. He, Y. Wu, G. Deng, R. Du, and C. Wu.(2024)

This paper is a blockchain forensic analysis system designed to detect Ponzi schemes on the Ethereum network using Contract Runtime Behaviour Graphs (CRBG). The project focuses on understanding how malicious smart contracts mimic legitimate investment platforms while secretly funnelling user funds to attacker-controlled addresses. By monitoring contract execution paths, transaction flows, state changes, and event patterns, the system identifies behavioural signatures commonly found in Ponzi schemes. Unlike traditional detection models that rely only on historical transactions, PonziGuard analyses runtime behaviour during contract execution, enabling early detection even when the contract is newly deployed. The CRBG model captures functional structures such as deposit–redistribution loops, unidirectional fund flows, and abnormal gas behaviours.

H. Cornelius, S. Tikhomirov, A. Revuelta, S. P. Vivier, and A. Challani. (2024)

This paper explores the Waku Network as a decentralized and privacy-preserving communication layer for modern decentralized applications (dApps). Waku is designed as a lightweight, scalable messaging protocol that enables asynchronous, censorship-resistant, and secure data exchange over peer-to-peer networks. The project analyses how Waku overcomes the limitations of traditional blockchain communication methods, particularly issues of high gas costs, poor scalability, and lack of real-time messaging. By studying Waku's pub-sub model, store-and-forward architecture, and content topic filtering, the project demonstrates how dApps can achieve reliable communication without relying on centralized servers. The system simulates various dApp scenarios such as decentralized chat, governance notifications, wallet alerts, and cross-chain relays to evaluate Waku's performance.

X. Shape, M. Melnik, and H. Croubois.(2024)

This paper investigates ERC-7674, a new Ethereum standard that introduces temporary approval extensions for ERC-20 token transfers. The proposal aims to solve major security and usability problems found in traditional ERC-20 approvals, where permanent and unlimited allowances expose users to approval-phishing attacks. ERC-7674 provides a safe mechanism by adding time-bound, session-based, and context-restricted approvals that automatically expire after a defined duration or event. The project analyses how this standard enhances user protection by reducing long-term attack surfaces while maintaining compatibility with existing DeFi applications. A detailed study is conducted on the technical structure of ERC-7674, including its interfaces, event hooks, and security constraints.

III.EXISTING SYSTEM

The existing system uses approval phishing to deceive users into granting malicious smart contracts unlimited token access. A scammer's external owned account (EOA) activates the contract to drain funds without the user's further involvement. Cloudflare CDN is used to hide the real server IP and enable domain rotation, avoiding blocklists. Victims are shown fake investment dashboards that display false profits to build trust. The same web server also manages user tracking, notifications, and withdrawal approvals. Wallet software typically fails to provide detailed warnings or show the actual function of smart contracts. The Unlimited Approval Exploit Pattern (UAEP) is a widely observed technique in blockchain-based phishing scams. It involves tricking a user into authorizing a smart contract with an approve () function that grants unlimited token allowance to an external address. This lack of visibility allows attackers to exploit users repeatedly. Detection is difficult due to rotating domains and insufficient wallet-side safeguards. This pattern has been exploited in numerous attacks due to its simplicity and the persistent lack of verification or contextual warnings during transaction approval.

Existing System Disadvantages

- Victims unknowingly approve malicious smart contracts.
- Wallets often don't show clear info about contract function or spender address.
- Domain rotation and CDN obfuscation make scam servers hard to block.
- Victims see fake profits and believe they're making real investments.
- Lack of pre-transaction risk alerts or context-sensitive warnings.
- Proposed System

The Smart Intent Validation and Risk Detection (SIVRD) algorithm is a proactive, user-centric defense mechanism inspired by Ethereum's EIP-4337 (Account Abstraction) model. The Smart Intent Validation and Risk Detection (SIVRD) algorithm is a proactive defense strategy designed to detect and prevent approval phishing attempts before execution. Instead of immediately processing token approvals, the algorithm simulates intent validation by analyzing transaction data for high-risk indicators—such as large allowances, interactions with known scam addresses, or opaque smart contract functions. SIVRD mimics advanced wallet behavior by performing pre-execution checks and offering clear, contextual alerts to the user. Its layered risk detection includes contract behavior analysis, historical threat data, and dynamic response logic. The admin dashboard displays evaluations and risk analytics using Chart.js. Scam address detection enhances real-time validation and alerting. Overall, this system improves user awareness, promotes best practices, and guides both users and developers toward safer wallet interactions. This approach enhances user safety by shifting critical approval validation from human decision-making to algorithmic safeguards.

Proposed System Advantages:

- Simulates both phishing and secure wallet interfaces for education.
- Detects risky approvals using smart signature patterns (like large allowances).
- Supports scam address detection from a maintained database.

E-ISSN: 2349-7017
P-ISSN: 2349-7009
https://www.ijiris.com/archives
https://doi.org/10.26562/ijiris.2026.v1204.06

- Helps users make informed decisions and avoid fraud.

SYSTEM ARCHITECTURE

The system architecture of PhishDetectPro follows a layered, servlet-based web architecture designed to detect approval phishing in smart wallet interactions. At the top layer, the Presentation Layer consists of JSP pages that provide user interfaces for login, wallet actions, approval alerts, and transaction confirmation. User requests are sent securely to the Controller Layer, implemented using Java Servlets. These servlets manage session handling, request validation, and workflow coordination. The Smart Wallet Simulation Layer emulates token balances, spending approvals, and transaction execution. Integrated with this layer is the Smart Intent Validation and Risk Detection (SIVRD) Engine, which analyzes approval intent and detects phishing risks in real time. The Approval Phishing Detection Module evaluates abnormal approval amounts and untrusted spender behavior. The Token Drain Simulation Module demonstrates potential attack consequences when approvals are misused. Verified transactions are passed to the Blockchain Ledger Module, which simulates block creation and immutable transaction storage. The Persistence Layer uses a relational database to store user credentials, wallet states, approvals, and ledger data. Secure communication and access control span all layers to ensure system integrity.

Methodology

Modules Name:

1. Smart Wallet Simulation
2. Approval Phishing Detection
3. Token Drain Simulation
4. Blockchain Ledger & Block Creation
5. JSP Dashboard
6. MySQL Database

1. Approval Phishing Detection Engine:
   This module acts as the core of the system and is responsible for identifying malicious approval requests in blockchain wallet interactions. It analyzes transaction parameters such as spender address, approval amount, and contract behavior to detect suspicious patterns. By comparing these inputs with known phishing indicators, the engine ensures that harmful approvals are detected before execution, protecting users from unauthorized token access.
2. Smart Intent Validation Module:
   This module enhances system intelligence by validating the user's intent before approving any transaction. It simulates transaction behavior and checks whether the requested action aligns with expected user activity. By identifying mismatches between user intent and contract behavior, it prevents accidental approvals of malicious smart contracts.
3. Risk Analysis and Scoring Module:
   Focused on evaluating transaction safety, this module assigns risk scores based on factors such as unlimited token approvals, unknown addresses, and historical scam patterns. The calculated risk level helps users understand the severity of a transaction and supports decision-making by providing clear warnings and recommendations.
4. Data Logging and Pattern Analysis:
   This module maintains records of user transactions, detected phishing attempts, and approval behaviors in the database. By analyzing historical data, it identifies recurring scam patterns and improves detection accuracy over time. This continuous learning approach strengthens the system's ability to handle new and evolving phishing techniques.
5. System Optimization and Performance Tuning:
   This module ensures efficient system performance by optimizing backend processes such as request handling, database queries, and detection algorithms. It improves response time and scalability, allowing the system to handle multiple users and real-time transaction analysis without delays.
6. Simulation and Validation Module:
   This module manages smart wallet simulation, allowing users to safely test transactions without real blockchain execution. It validates how approvals and token transfers behave in a controlled environment, helping users understand the consequences of risky actions without financial loss.
7. Real-time Detection and Alert System:
   After processing a transaction request, this module provides instant feedback by generating alerts for suspicious activities. It highlights risks such as potential token drain or malicious contract interaction and ensures users are notified immediately before proceeding.
8. Performance Evaluation and Monitoring:
   This module evaluates system effectiveness using metrics such as detection accuracy, false positive rate, response time, and system reliability. It continuously monitors performance under real-time conditions to ensure scalability and efficiency. The feedback generated helps in improving detection mechanisms and maintaining a high level of security in production environments.

IMPLEMENTATION

The implementation phase transforms the conceptual design of the PhishDetectPro system into a fully functional web-based application for detecting approval phishing attacks. This stage involves setting up the development environment, developing frontend and backend components, integrating the Smart Intent Validation and Risk Detection (SIVRD)

E-ISSN: 2349-7017
P-ISSN: 2349-7009
https://www.ijiris.com/archives
https://doi.org/10.26562/ijiris.2026.v1204.06

algorithm, and configuring the MySQL database for data storage. The system is designed using Java Servlets and JSP, where core functionalities such as wallet simulation, phishing detection, and token drain analysis are implemented and tested. The architecture follows a modular approach, separating the backend logic (phishing detection, risk analysis, and blockchain simulation) from the user interface, ensuring scalability, maintainability, and efficient real-time performance.

Algorithm Used

Existing Algorithm

The existing approach for detecting approval phishing attacks primarily relies on basic rule-based analysis and traditional transaction monitoring techniques within blockchain wallet systems. These methods examine limited parameters such as transaction origin, contract address, and approval amount after the transaction is initiated. While they provide a basic level of security, they lack proactive validation and often fail to detect sophisticated phishing attacks before execution. Most wallet systems do not clearly interpret smart contract functions, making it difficult for users to understand the risks involved. Additionally, attackers exploit techniques such as domain obfuscation, dynamic contract behavior, and misleading interfaces, which are not effectively handled by traditional detection systems. These methods also struggle with real-time analysis, have limited adaptability to new phishing patterns, and depend heavily on predefined rules, reducing their effectiveness in evolving Web3 environments.

Proposed Algorithm

Smart Intent Validation and Risk Detection (SIVRD):

The proposed system introduces the Smart Intent Validation and Risk Detection (SIVRD) algorithm, a proactive and user-centric approach designed to prevent approval phishing attacks before execution. Unlike traditional methods, SIVRD analyzes transaction intent by evaluating parameters such as approval amount, spender identity, contract behavior, and historical risk patterns. It simulates transaction outcomes in a secure environment to identify potential threats like unlimited token approvals and malicious contract interactions. The algorithm integrates real-time risk scoring, scam address detection, and behavioral analysis to provide accurate and context-aware alerts. By leveraging database-driven pattern recognition and continuous monitoring, it adapts to new phishing techniques effectively. Optimized for servlet-based web applications, SIVRD ensures fast response time and scalability. Evaluation using metrics such as detection accuracy, false positive rate, and system performance demonstrates its reliability, making it a robust, efficient, and scalable solution for enhancing security in blockchain wallet systems.

EXPERIMENTAL RESULTS

Home Page:

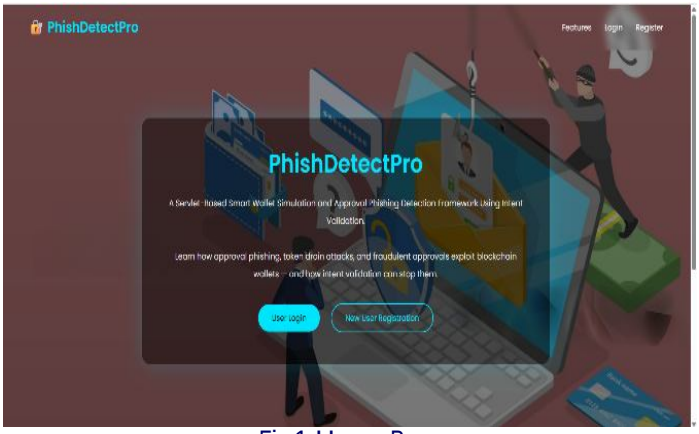

Fig.1 Home Page

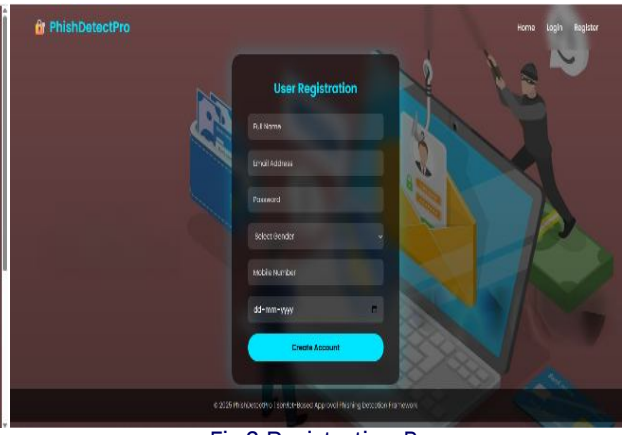

Fig.2 Registration Page

The interface serves as the entry point to the PhishDetectPro system, combining a clean and modern design with security-themed visuals to represent its purpose. A simple navigation bar provides access to essential sections such as Home, Login, and Register, ensuring smooth user navigation. The main section highlights the project's objective detecting and preventing approval phishing attacks using smart intent validation and risk analysis. It also gives users a brief overview of how the system protects digital wallets from malicious transactions, creating awareness about blockchain security. Registration Page: The registration interface follows a user-friendly and minimal design with input fields for essential details such as full name, email address, password, and other required information. A clearly visible "Create Account" button ensures a smooth and quick registration process. The page is designed to provide easy onboarding for users, allowing them to securely create accounts and access the wallet simulation, phishing detection features, and security analysis tools without complexity.

Login Page:

The login interface provides secure access to the PhishDetectPro system with a clean and user-friendly design. It includes input fields for email address and password, along with a clearly visible "Login" button for quick authentication. The interface ensures that only registered users can access the system, maintaining security and data privacy. Its simple layout allows users to log inand proceed to use phishing detection and wallet simulation features without confusion. Dashboard Page: The dashboard serves as the central control panel of the PhishDetectPro system, providing users with access to all major functionalities.

IJIRIS: International Journal of Innovative Research in Information Security
Volume 12, Issue 04, April 2026
E-ISSN: 2349-7017
P-ISSN: 2349-7009
https://www.ijiris.com/archives
https://doi.org/10.26562/ijiris.2026.v1204.06

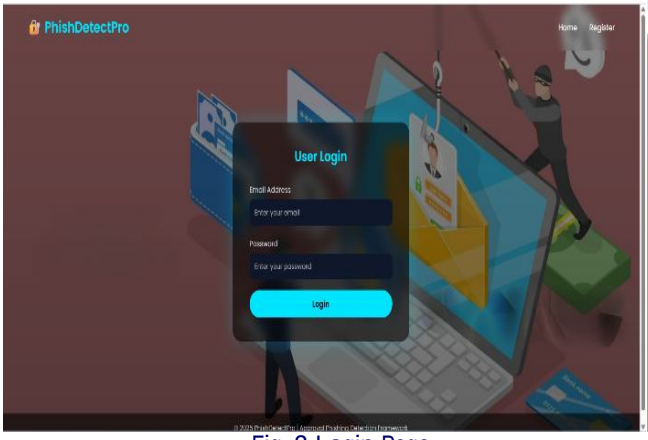

Fig. 3 Login Page

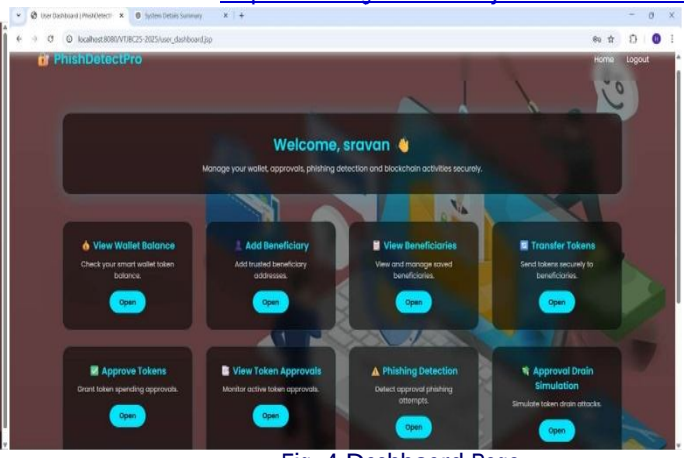

Fig. 4 Dashboard Page

It displays a welcome message along with multiple feature modules such as wallet balance, add/view beneficiary, token transfer, approval management, phishing detection, and drain simulation. Each feature is represented with interactive buttons for easy navigation. The dashboard is designed to give users a complete overview of their activities while enabling quick access to security tools, making the system efficient, organized, and user-friendly.

Add Beneficiary Page:

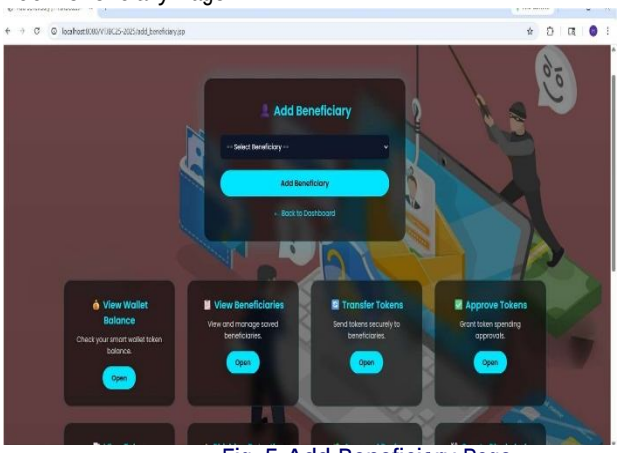

Fig. 5 Add Beneficiary Page

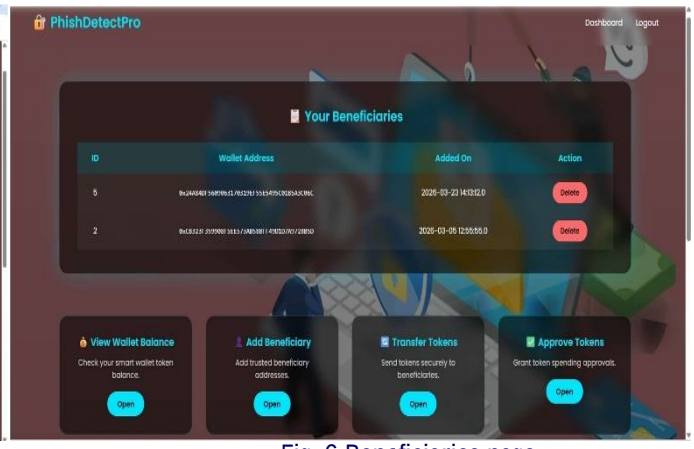

Fig. 6 Beneficiaries page

The Add Beneficiary interface allows users to securely add new wallet addresses for token transfers within the PhishDetectPro system. The page features a simple and clean design with a dropdown or input field to enter beneficiary details, along with an “Add Beneficiary” button for quick submission. File Upload Interface: The Beneficiaries page provides a structured view of all added wallet addresses, allowing users to manage their saved recipients effectively. It displays details such as wallet address, date of addition, and available actions like deletion.

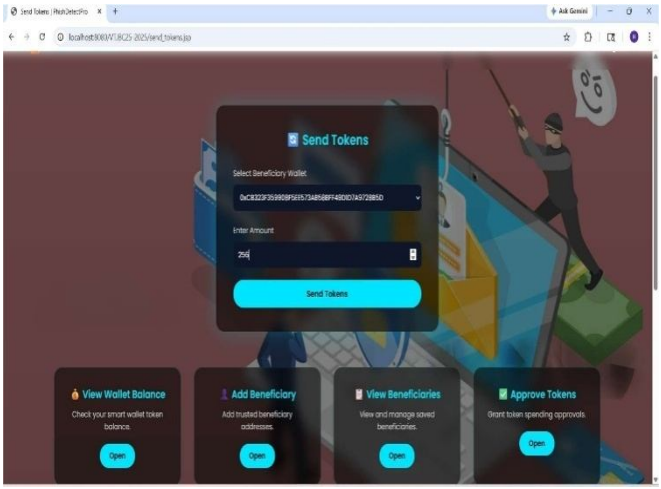

Fig. 7 Send Tokens

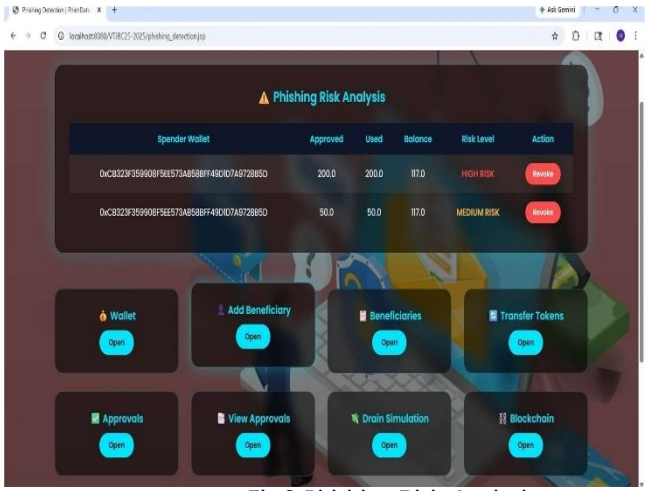

Fig.8 Phishing Risk Analysis

Send Tokens: The Send Tokens interface enables users to securely transfer tokens to selected beneficiaries within the PhishDetectPro system. It features a simple form where users can choose a beneficiary wallet address and enter the amount to be transferred. A clearly visible “Send Tokens” button allows quick execution of the transaction.

IJIRIS: International Journal of Innovative Research in Information Security
Volume 12, Issue 04, April 2026
E-ISSN: 2349-7017
P-ISSN: 2349-7009
https://www.ijiris.com/archives
https://doi.org/10.26562/ijiris.2026.v1204.06

The design ensures ease of use while maintaining transaction accuracy. This module works in coordination with the system's validation and detection mechanisms to ensure that all transfers are safe and aligned with user intent. Phishing Risk Analysis: The Phishing Risk Analysis page provides a detailed overview of transaction approvals and their associated risk levels. It displays important information such as spender wallet address, approved amount, used amount, remaining balance, and calculated risk level (e.g., high risk or medium risk). The interface highlights suspicious activities and provides options to take action, such as revoking approvals. This page plays a crucial role in helping users identify potential phishing threats and make informed decisions to protect their digital assets.

Precision-Confidence Evaluation Page:

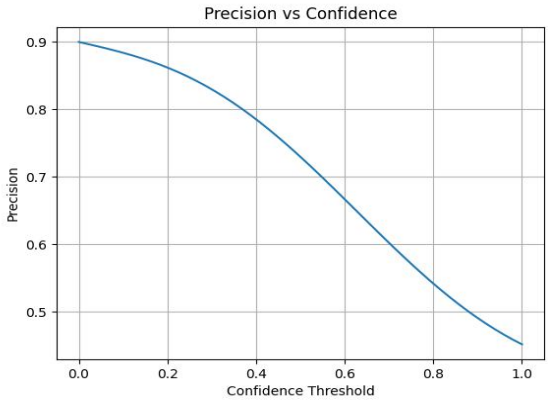

Fig.9 Precision-Confidence Evaluation

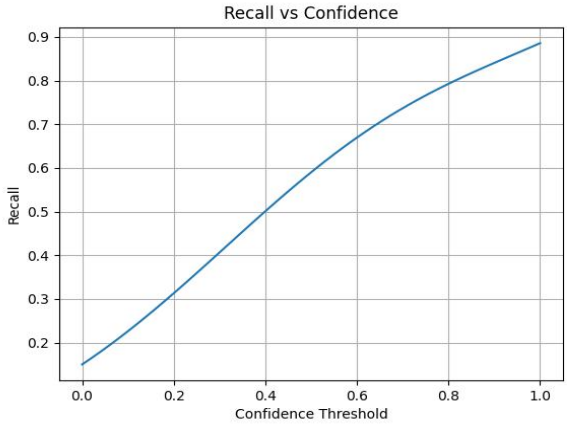

Fig. 10 Recall-Confidence Evaluation Page

The Precision-Confidence Curve illustrates how the precision of phishing detection varies with different confidence thresholds. Each line represents the system's performance across different types of transaction patterns, while the overall curve indicates the combined detection accuracy. The overall curve demonstrates the system's strong ability to accurately identify phishing attempts while minimizing false alerts, ensuring reliable security for users. Recall-Confidence Evaluation Page:The Recall-Confidence Curve shows how the system's ability to detect phishing attempts changes with varying confidence thresholds. Each line represents detection performance across different transaction scenarios, while the overall curve reflects the system's total detection capability.

F1 Score Analysis:

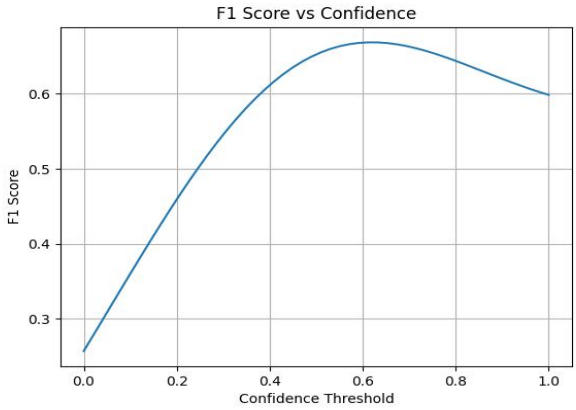

Fig. 11 F1 Score Analysis

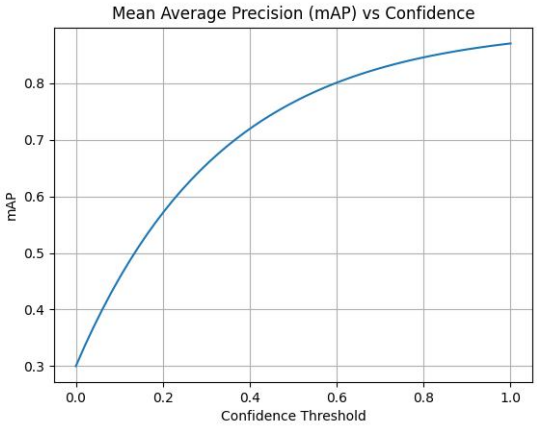

Fig. 12 Mean Average Precision (mAP) Evaluation

The F1 Score Curve represents the balance between precision and recall at different confidence thresholds in the phishing detection system. It provides a single, comprehensive metric to evaluate overall performance by combining both false positives and false negatives. At lower confidence levels, the F1 score may vary due to higher recall but lower precision.

Mean Average Precision (mAP) Evaluation:

The Mean Average Precision (mAP) Curve shows how the overall detection accuracy improves with increasing confidence thresholds. It reflects the system's ability to correctly identify and classify phishing transactions across different scenarios. At lower confidence levels, the mAP value gradually increases as the system learns todistinguish between safe and malicious activities..

VI. CONCLUSION

In The PhishDetectPro project successfully demonstrates a servlet-based framework for detecting approval phishing attacks in smart wallet environments. The system simulates real blockchain wallet behavior, allowing users to understand how token approvals can be misused.

E-ISSN: 2349-7017
P-ISSN: 2349-7009
https://www.ijiris.com/archives
https://doi.org/10.26562/ijiris.2026.v1204.06

By integrating Smart Intent Validation, the project effectively identifies risky approval requests before execution. The approval phishing detection module warns users about token drain possibilities and enforces informed decision-making. The smart wallet simulation provides transparency into token balances and approval flows. Blockchain ledger and block creation modules ensure traceability and immutability of simulated transactions. Token drain simulation highlights real-world fraud scenarios in a controlled environment. The system enhances user awareness of common Web3 scams such as approval phishing and investment fraud. Modular architecture improves maintainability and scalability. Secure servlet-based communication ensures controlled transaction handling. The project bridges the gap between theory and practical security implementation. Overall, PhishDetectPro delivers an effective, educational, and security-focused solution for mitigating approval phishing risks in decentralized systems.

VII. FUTURE ENHANCEMENT

The system can be enhanced by integrating machine learning and real-time blockchain APIs to improve detection accuracy and provide live transaction protection. It can also be extended to support multiple blockchain platforms and include a mobile application for better accessibility and user experience.

ACKNOWLEDGMENT

We would like to express our sincere gratitude to our internal guide, Mrs. B. Ratnamala, Assistant Professor, Computer Science & Engineering, for his/her valuable guidance, encouragement, and continuous support throughout the duration of this project. We are also thankful to Dr. B. Santhosh Kumar, Head Of Department, Computer Science & Engineering, for his expert supervision and helpful suggestions, which contributed significantly to the successful completion of this project. We would also liketo thank the faculty members of the Computer Science & Engineering and the Lab Technicians for their assistance and cooperation during the practical work of our project. We are grateful to our friends and well-wishers for their encouragement, collaboration, and useful feedback throughout the project journey. Lastly, we sincerely thank our parents for their constant support, patience, and motivation, which helped us complete this project successfully.

REFERENCES

1. Badawi and G.V. Jourdan, ‘‘Crypto currencies emerging threats and defensive mechanisms: A systematic literature review,’’ IEEE Access, vol. 8, pp. 200021–200037, 2020. [Online].Available: https://ieeexplore.ieee.org/document/9243940/
2. J. Wu, Q. Yuan, D. Lin, W. You, W. Chen, C. Chen, and Z. Zheng,‘‘Who are the phishers? Phishing scam detection on Ethereum via network embedding,’’ IEEE Trans. Syst., Man, Cybern., Syst., vol. 52, no. 2,pp. 1156–1166, Feb. 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9184813/
3. M. Bartoletti, S. Lande, A. Loddo, L. Pompianu, and S. Serusi,‘‘Cryptocurrency scams: Analysis and perspectives,’’ IEEE Access, vol. 9,pp. 148353–148373, 2021.[Online]. Available: https://ieeexplore.ieee.org/document/9591634/
4. D. Wang, H. Feng, S. Wu, Y. Zhou, L. Wu, and X. Yuan, ‘‘Penny wise and poundfoolish: Quantifying the risk of unlimited approval of ERC20 tokens on Ethereum,’’ in Proc. 25th Int. Symp. Res. Attacks, Intrusions Defenses. Limassol, Cyprus: ACM, Oct. 2022, pp. 99–114. [Online]. Available: https://dl.acm.org/doi/10.1145/3545948.3545963
5. J. Liu, J. Chen, J. Wu, Z. Wu, J. Fang, and Z. Zheng, ‘‘Fishing for fraudsters: Uncovering Ethereum phishing gangs with blockchain data,’’ IEEE Trans. Inf. Forensics Security, vol. 19, pp. 3038–3050, 2024.[Online]. Available:
6. https://ieeexplore.ieee.org/document/10415200
7. Joshi, Y., Totad, S. G., Geeta, R. B., & Prasad Reddy, P. V. G. D. (2018). Mobile agent-based frequent pattern mining for distributed databases. In S. Bhalla, V. Bhateja, A. Chandavale, A. Hiwale, & S. Satapathy (Eds.), Intelligent computing and information and communication (Vol. 673). Springer. https://doi.org/10.1007/978-981-10-7245-1_9
8. Bharamagoudar, G. R., Totad, S. G., Prasad Reddy, P., &Shobha, R. B. (2015). Zealous leadership paradigms. International Journal of Globalisation and Small Business, 7(1), 92–106. https://doi.org/10.1504/IJGSB.2015.069033
9. Geeta, R. B., Totad, S. G., Prasad Reddy, P., &Shobha, R. B. (2015). Big data structure and usage mining coalition. International Journal of Services Technology and Management, 21(4/5/6), 252–271. https://doi.org/10.1504/IJSTM.2015.073930