International Journal of Engineering Research and Science & Technology

PHISHDETECTPRO: A SERVLET-BASED SMART WALLET SIMULATION AND APPROVAL PHISHING DETECTION FRAMEWORK USING INTENT VALIDATION

Dr. NAVEEN REDDY NEMILI[1], SHIVAGOURISHETTI NAVEENA[2], VANGA VINEETH REDDY[3], YELLAGONDA ROHIT REDDY[4], KOMATI SHANVI[5]

[1]Assistant Professor, Dept. of CSE, Guru Nanak Institutions Technical Campus**,** Ibrahimpatnam, Telangana, India

[2, 3, 4, 5] B.Tech, Dept. of CSE, Guru Nanak Institutions Technical Campus**,** Ibrahimpatnam, Telangana, India

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 block listing. 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 analyse these attacks through a Servlet-JSP-based web application, evaluate wallet vulnerabilities, detect scam contract behaviour, 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 secure wallet choices.

Index Terms – Blockchain security, approval phishing, smart contracts, externally owned account (EOA), cryptocurrency wallet security, account abstraction, EIP-4337, phishing detection, decentralized applications (DApps), blockchain fraud

detection, token approval, smart wallet, cybersecurity.

I. INTRODUCTION

Blockchain-based cryptocurrencies differ from traditional financial systems in many aspects. On one hand, they offer a much greater level of transparency and security due to their distributed nature and advanced cryptography. On the other hand, however, the complexity and lack of a central, trusted authority overseeing transactions introduce new threats to users. The threats are further exacerbated by the continuously increasing popularity and value of cryptocurrency markets. This trend attracts both less technologically aware users who wish to invest in new technologies– and malicious actors, who seek to profit illegally through various fraudulent techniques. Unsurprisingly, users from the first group frequently fall victim to those from the second. What worsens the situation is the continuous evolution of blockchain technology. A significant functional advancement was the introduction of smart contracts, which allowed users to create arbitrary logic within a blockchain system. A smart contract is fully defined by its source code and operates autonomously once deployed to the blockchain. Subsequently, any user of the platform can execute actions defined within that contract and obtain predictable results. This feature of blockchain technology offers limitless possibilities but also increases its complexity. Among the many associated risks, one significant issue is users' understanding of the source code and the possible outcomes of its execution. For instance, many blockchain technologies that support smart contracts have standardized so-called tokens. In Ethereum, these are known as ERC-20; in Tron, TRC-20; in Binance Smart Chain, BSC-20; and so on. Simply put, tokens are like goods that can be owned and transferred meaning they can be bought, sold, or even stolen. Some tokens function as new currencies. Although they are stored on the underlying blockchain system, these new currencies have very little in common with the blockchain's native cryptocurrency. One critical distinction, especially relevant to this paper, is that a blockchain's native cryptocurrency coins can be controlled solely by the owner, who is authorized through the proper private key and cryptography– a condition that does not necessarily apply to tokens. In the case of tokens, this restriction is considerably weaker. Token smart contracts include an approve function, which allows users to authorize another party to

transfer their tokens without requiring further interaction from the owner. While this delegation pattern introduces numerous new possibilities, it also increases the risk of unintentionally granting malicious actors the ability to steal tokens. This article demonstrates how this feature is exploited by real-world criminals to steal millions of dollars every month from users around the world. Four different scamming groups were monitored between November 2023 and September 2024. During this period, the author gained detailed knowledge of how the scam operates—how the scammers are organized, what tools they use, the typical scenario, as well as information about the victims (nationality, wallet addresses, and lost amounts) and the scammers (IP addresses, usernames, etc.). In addition to the results of the monitored scam, the author conducted a thorough analysis of wallets available on the market, evaluating their maturity and security in relation to approval handling functions. Finally, this analysis led to the identification of new security controls and the creation of recommendations, including ways to avoid such risks. Blockchain frauds are widely studied by security researchers around the world. This includes meta-analyses such as blockchain transaction and smart contract data mining–aiming to identify patterns in smart contract and transaction details, as in works like as well as numerous case studies presenting analyses of specific attacks. This article focuses on the topic of tokens, which are new types of assets established by smart contracts built on top of blockchain technology. Tokens can be used in various ways, but they are typically created to reflect real-world value owned by specific users. In particular, tokens often serve as the foundation for new cryptocurrencies. Examples include well-known ERC-20 tokens such as wETH, Tether, BNB, and USDT. For additional tokens, please refer to analytics platforms related to the specific blockchain technology, such as Etherscan for Ethereum). From a technical perspective, there are many possible attack vectors related to token schemes and implementations. In this model, the attacker, using a vulnerability in existing and trusted dApps, can force the smart contract to transfer funds to the attacker's address. Most interesting from the perspective of this paper, also called approval phishing, attackers use social engineering (usually investment frauds or pig butchering) to force investors to accept an approval transaction. While this transaction itself is inexpensive, requiring only gas fees, it

secretly approves the attacker's wallet to transfer the victim's assets.

II. RELATED WORK

A. Fishing for fraudsters: Uncovering Ethereum phishing gangs with blockchain data

This project 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. The system further analyses wallet vulnerabilities, highlighting how poor notification design and unclear approval warnings make users susceptible. Using blockchain analytics, the project tracks fund drainage patterns, batching behaviour, gas usage anomalies, and cross-chain swaps performed by attackers. Visual dashboards help represent the flow of stolen assets across networks and exchanges. Comparative evaluation of popular Ethereum wallets is conducted to assess how well they detect or prevent unsafe approvals. The insights gained are used to design improved wallet mechanisms inspired by account abstraction (EIP-4337). Finally, the project produces a recommendation framework to educate users, enhance wallet security, and enable early detection of phishing gang activities using blockchain intelligence.

B. PonziGuard: Detecting Ponzi schemes on Ethereum with contract runtime behaviour graph (CRBG)

PonziGuard 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. The system processes on-chain data, builds graph representations, and classifies suspicious contracts through behavioural clustering. A Servlet–JSP web portal is implemented to visualize graphs, show incoming and outgoing fund paths, and provide real-time contract risk scores. PonziGuard also studies scammer wallet networks by linking addresses involved in creator–withdrawal–redistribution cycles. The system highlights deceptive contract features, including hidden owner privileges, hardcoded withdrawal patterns, and misleading "profit-sharing" logic. Comparative evaluation with existing detection methods shows improved accuracy and early-warning capability. The project ultimately helps users, auditors, and researchers understand Ponzi contract operations, detect fraud before victims lose funds, and strengthen smart contract security across the Ethereum ecosystem.

C. The waku network as infrastructure for dApps

This project 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. The project also examines Waku's underlying protocols (Waku Relay, Waku

Store, Waku Light Push, and Waku Filter) to show how they provide both full-node and light-client support. A Servlet–JSP based dashboard is implemented to visualize message propagation, node connectivity, and network latency metrics. Privacy mechanisms like message encryption, metadata protection, and spam-resistant routing are analysed to highlight Waku's advantages over traditional messaging layers. The project further compares Waku with Whisper, Libp2p PubSub, and centralized messaging APIs. Performance tests focus on reliability under network congestion, bandwidth efficiency, and fault tolerance. Ultimately, the project demonstrates how Waku can serve as a robust infrastructure layer that empowers decentralized applications with secure, scalable, and censorship-resistant communication.

D. ERC-7674: Temporary Approval Extension for ERC-20, Standard 7674, Ethereum Improvement Proposals

This project 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. Using Solidity, an experimental smart contract is developed to demonstrate how temporary approvals are granted, monitored, and revoked on-chain. A Servlet–JSP simulation dashboard is implemented to visualize approval timelines, expiry behaviour, transfer attempts, and security alerts. The system compares ERC-20, Permit2, and ERC-7674 to highlight improvements in risk mitigation and transaction safety. Runtime testing is performed to evaluate gas efficiency, UX enhancements, and backward compatibility with dApps. Attack simulations—such as approval phishing, malicious contract drainage, and long-term allowance exploits—are used to measure resilience gained through temporary approvals. The project also explores integration possibilities with account

abstraction (EIP-4337) to enable smart wallets with automated approval expiry. Overall, the project demonstrates how ERC-7674 provides a practical, secure, and user-centric evolution of token approval mechanisms on Ethereum.

III. PROPOSED METHODOLOGY

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.

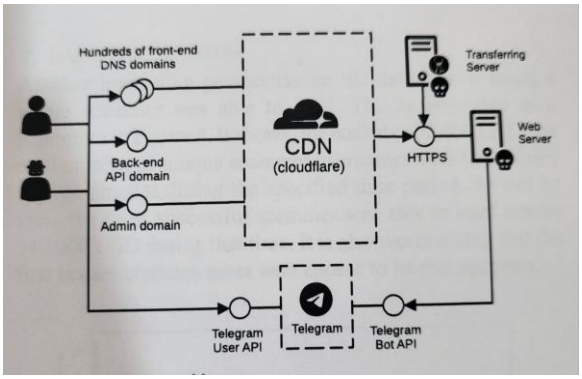
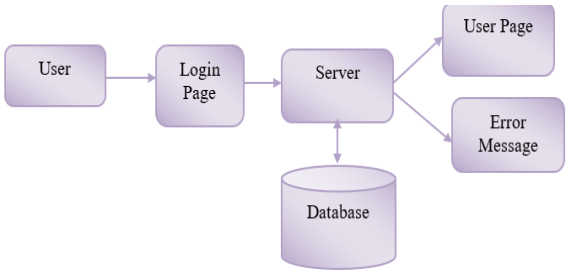


Fig. 1: System Overview

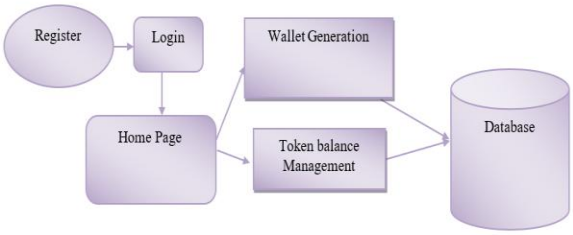
1. User Interface Design

To connect with server user must give their username and password then only they can able to connect the server. If the user already exits directly can login into the server else user must register their details such as username, password, Email id, City and Country into the server. Database will create the account for the entire user to maintain data and download rate. Name will be set as user id. Logging in is usually used to enter a specific page. It will search the query and display the query.



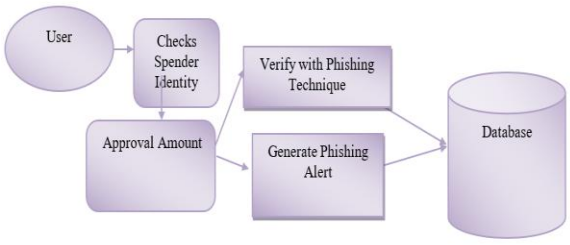
2. Smart Wallet Simulation

The Smart Wallet Simulation Module emulates the behavior of a real blockchain-based wallet within a Java Servlet environment. Instead of deploying actual smart contracts, this module reproduces core wallet functionalities such as wallet address generation, token balance management, approval granting, and token transfers using server-side logic and persistent storage. It models ERC-20–style mechanisms where token owners can approve third-party spenders to use their tokens. By maintaining wallet state in a backend database and enforcing transaction rules through servlets, this module enables controlled experimentation with wallet operations, making it ideal for demonstrating security vulnerabilities like approval misuse without relying on real blockchain infrastructure.



3. Approval Phishing Detection

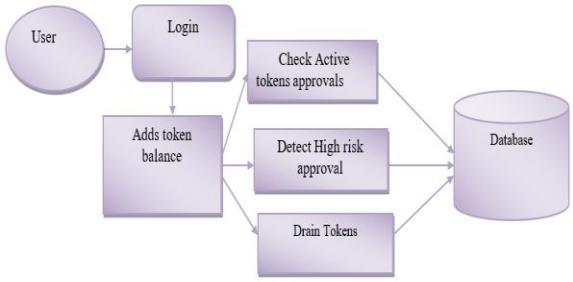
The Approval Phishing Detection Module is responsible for identifying malicious approval attempts that could lead to unauthorized token access. This module analyzes approval parameters such as spender identity, approval amount, approval frequency, and historical wallet behavior before allowing an approval to proceed. It compares these parameters against predefined risk indicators that resemble real-world approval phishing techniques used in DeFi scams. When a suspicious pattern is detected—such as unusually large approval limits or unknown spender addresses—the module generates a phishing alert and prompts the user with a security warning. This proactive detection mechanism highlights how approval phishing exploits user trust and demonstrates the importance of pre-approval validation in decentralized systems.



4. Token Drain Simulation

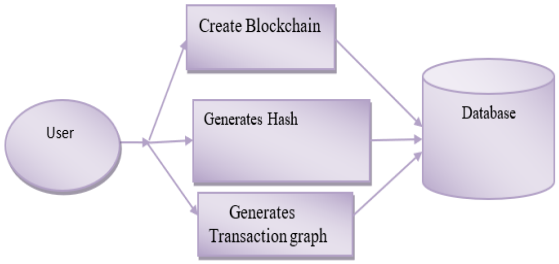
The Token Drain Simulation Module demonstrates the real impact of approval

phishing by simulating how attackers exploit active token approvals. Once a high-risk or malicious approval exists, this module uses the approved permissions to drain tokens from the user's wallet without further user consent, closely replicating real blockchain attack scenarios. The simulation is executed within a safe, controlled environment where token balances are adjusted server-side to reflect unauthorized transfers. By visualizing token loss caused by excessive or risky approvals, this module effectively educates users on the consequences of approval phishing and reinforces the need for careful permission management.



5. Blockchain Ledger & Block Creation

The Blockchain Ledger and Block Creation Module emulates blockchain data structures by organizing wallet transactions into sequential blocks. Each block contains transaction details, timestamps, cryptographic hashes, and a reference to the previous block, ensuring data integrity and immutability within the simulated environment. Although implemented off-chain using Java and database storage, this module closely follows blockchain principles to maintain a tamper-evident transaction history. It allows users to visualize how approvals, transfers, and drain events are recorded on a blockchain, supporting transparency, auditability, and trust. This module strengthens the realism of the framework by bridging conceptual blockchain mechanics with a servlet-based implementation.



6. JSP Dashboard

Displays the transaction data along with details and timestamps. It also shows the transaction status, verification results, pending registration details, and overall project modules. The dashboard acts as the user interface for monitoring stored data, status, and user details etc.

7. MySQL Database

The MySQL database is used to maintain user details, device metadata, and other non-critical information that supports the

system's operation. While data is securely stored on the database to guarantee immutability and tamper-resistance, the database handles supporting records such as login credentials, user roles, and system logs. This hybrid storage approach ensures both efficient system performance and strong data security by separating critical data from auxiliary information.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

The experimental results show that the proposed PhishDetectPro framework effectively detects approval phishing attacks by analyzing smart contract approval requests and validating user intent. The Servlet-JSP-based simulation successfully identifies malicious contract behavior and highlights vulnerabilities in existing cryptocurrency wallets. The integration of EIP-4337-inspired smart wallet features enhances transaction security and helps prevent unauthorized token approvals. Overall, the framework provides an effective solution for phishing awareness, secure wallet evaluation, and blockchain transaction protection.

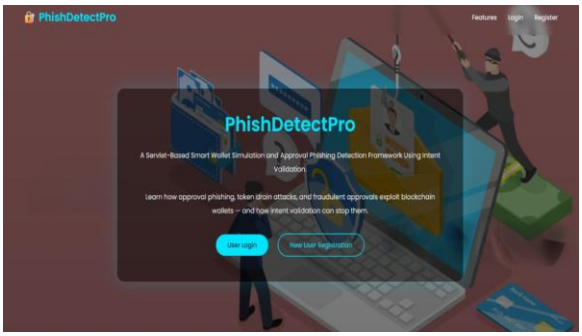


Fig. 2: Home Page

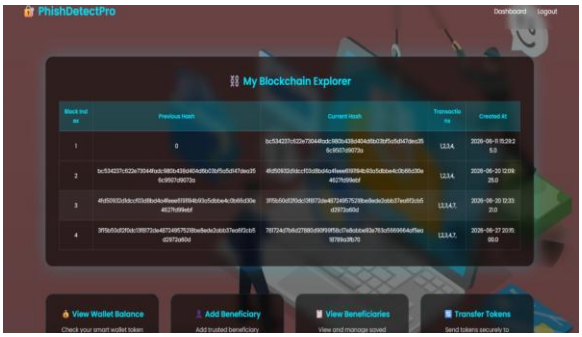


Fig. 3: Blockchain Explorer

Fig. 4: Transaction Graph

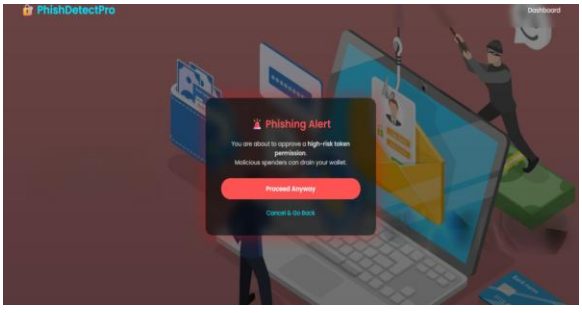


Fig. 5: Phishing Alert

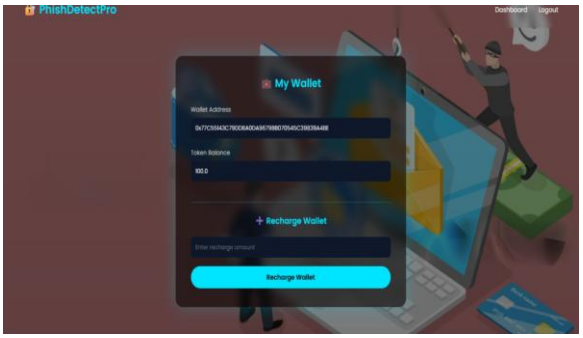


Fig. 6: My wallet

V. CONCLUSION

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. 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.

REFERENCES

[1] Badawi and G.-V. Jourdan, ‘‘Cryptocurrencies 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: https://ieeexplore.ieee.org/document/10415200

[6] Y. Zhang, W. Yu, Z. Li, S. Raza, and H. Cao, ‘‘Detecting Ethereum Ponzi schemes based on improved LightGBM algorithm,’’ IEEE Trans. Computat. Social Syst., vol. 9, no. 2, pp. 624–637, Apr. 2022. [Online].Available: https://ieeexplore.ieee.org/document/9479748/

[7] R. Liang, J. Chen, K. He, Y. Wu, G. Deng, R. Du, and C. Wu,‘‘PonziGuard: Detecting Ponzi schemes on Ethereum with contract runtime behavior graph (CRBG),’’ in Proc. IEEE/ACM 46th Int. Conf.Softw. Eng., Lisbon, Portugal, Feb. 2024, pp. 1–12. [Online]. Available: https://dl.acm.org/doi/10.1145/3597503.3623318

[8] M.Bartoletti, S. Carta, T. Cimoli, and R. Saia, ‘‘Dissecting Ponzi schemes on Ethereum: Identification, analysis, and impact,’’ Future Gener. Comput.Syst., vol. 102, pp. 259–277, Jan. 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X18301407

[9] A. Holub and J. O’Connor, ‘‘COINHOARDER: Tracking a Ukrainian Bitcoin phishing ring DNS style,’’ in Proc. APWG Symp. Electron. Crime Res. (eCrime), May 2018, pp. 1–5. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8376207/

[10] E. Badawi, G.-V. Jourdan, G. Bochmann, and I.-V. Onut,‘‘An automatic detection and analysis of the Bitcoin generator scam,’’in Proc. IEEE Eur. Symp. Secur. Privacy Workshops (EuroS&PW),Sep. 2020, pp. 407–416. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9229769/.

AUTHORS Profile

Dr. NAVEEN REDDY NEMILI has received his MCA from Osmania University, Hyderabad in 2010, M.Tech degree in Computer science from JNTUH, Hyderabad in 2014 and Ph.D degree awarded in Computer Science from SSSUTMS, Madhya Pradesh in 2024 He is dedicated to teaching field from the last 12 years. His research area Cloud Computing. At present he is working as Assistant Professor in CSE Department at Guru Nanak Institutions Technical Campus, Ibrahimpatnam, Telangana, India.

Ms. SHIVAGOURI SHETTI NAVEENA pursuing B.Tech degree in Computer Science and Engineering at Guru Nanak Institutions Technical Campus affiliated to JNTUH in 2022-2026.

Mr. VANGA VINEETH REDDY pursuing B.Tech degree in Computer Science and Engineering at Guru Nanak Institutions Technical Campus affiliated to JNTUH in 2022-2026.

Mr. YELLAGONDA ROHIT REDDY pursuing B.Tech degree in Computer Science and Engineering at Guru Nanak Institutions Technical Campus affiliated to JNTUH in 2022-2026.

Ms. KOMATI SHANVI pursuing B.Tech degree in Computer Science and Engineering at Guru Nanak Institutions Technical Campus affiliated to JNTUH in 2022-2026.