OPEN ACCESS

Kashf Journal of Multidisciplinary Research
Vol: 02 - Issue 4 (2025)
P-ISSN: 3007-1992 E-ISSN: 3007-200X
https://kjmr.com.pk

Ethereum Hidden Dangers: Ponzi Scheme Detection in Smart Contracts Using Sourcep

Noor ul Ain Afzal
Department of Computer Science, NFC IET, Multan, Pakistan.
Muhammad Kamran Abid*
Department of Computer Science, Emerson University, Multan, Pakistan.
Muhammad Fuzail
Department of Computer Science, NFC IET, Multan, Pakistan.
Naeem Aslam
Department of Computer Science, NFC IET, Multan, Pakistan.
Nasir Umer
Department of Computer Science, NFC IET, Multan, Pakistan.

****Corresponding author: Muhammad Kamran Abid (**kamranabidhiraj@gmail.com***)*

Article Info

Abstract

Ponzi schemes have surfaced on the Ethereum platform as blockchain technology continues to gain traction. Using smart contracts, these schemes, also referred to as smart Ponzi schemes, have caused significant financial losses and adverse effects. Byte code features, op code characteristics, account qualities, and smart contract transaction behavior are the main focus areas for current Ethereum smart Ponzi scheme detection techniques. However, these methods often do not record the behavioral features of the Ponzi scheme, resulting in high false alarm rates and poor identification accuracy. In this study, we provide the source P. Source P is a unique way of knowing intelligent Ponzi schemes on the Ethereum platform, passed by dataflow. Using the intelligent contract's source code as a function eliminates the difficulty of collecting data and extracting functions from available identification methods. In particular, we convert the code into statistical flow diagrams, apply educated models, and use code representations to create classification models for the detection of Ponzi schemes. Experimental results show that SourceP outperforms cutting-edge technology in terms of sustainability and effectiveness, achieving an F1 score of 92.4% and a recall of 90.1% in Ethereum's smart Ponzi schema detection. Ponzi, Blockchain, Source Code, Intelligent Contracts.

Keywords:
Smart Ponzi Schemes, Ethereum Blockchain, Source Code Analysis, Intelligent Contract Detection.

1. Introduction

The swift embrace of blockchain technology, especially via Ethereum, catalyzed creative financial uses but also introduced enormous new dangers, particularly smart Ponzi schemes. These schemes utilize the blockchain's critical anonymity and decentralization features, causing massive financial damage to naive investors. The current detection tools analyze byte code and transaction behaviors, but they lack sufficient interpretability along with the sustainability needed to stop these types of fraudulent activities effectively [1]. SourceP employs three detection improvement mechanisms consisting of source code analysis in conjunction with pre-training models and data flow technology. SourceP enhances both data collection capabilities and analysis interpretation to improve detection accuracy in the system [2]. Smart contracts provide fraudsters an opportunity to operate Ponzi schemes through their unclear nature because investors fall victim to deceptive operations [3]. These schemes use the combination of people's lack of crypto knowledge and irrational crypto hype to convince investors about non-existent risks while promising eye-catching returns. The Ethereum ecosystem faces danger because it lacks effective detection systems to fight advanced scams that protect investors' safety and the stability of their investments [4]. The research aims to understand Ponzi scheme dangers in Ethereum smart contracts so it can put forward SourceP as a strategic protection framework that prevents these risks. The enhanced smart contract security tactics present substantial concerns because they might prevent future abusive schemes from entering the Ethereum network [5,6]. Moreover, SourceP demonstrates the fundamental work done on SourceP, further highlighting the importance of this research in the analysis.

1.1 Overview of Ethereum and Intelligent Contracts

An innovative self-executing contract can be established on the Ethereum blockchain by creating intelligent contracts and coding their respective clauses. Ethereum not only helps its users build but also run their smart contracts. The self-executing agreements are revolutionizing the functioning of many institutions and allow for the emergence of intermediaries, free, decentralized applications Apps. Industries like finance and even supply chain management are being changed by it, and many more are bound to follow. Sadly, alongside the benefits also come a fair share of challenges; further complications revolve around the complexities of blockchain enabling numerous systems to be misused for fraud[7], [8]. Such schemes use smart contracts for the automatic collection of funds and distribution which fools users into believing that they are entering a golden opportunity. So far, the focus of detecting these contracts has been on the transaction behavior of the byte code, which is quite hollow in its interpretation[9]. Still, SourceP provides an improvement by using the smart contract source code, thus making the methods more intuitive and more basic to grasp and allowing monitoring of potential Ponzi schemes in the Ethereum ecosystem (10).The illustration of the smart contract development process in further underscores the importance of vigilance in this rapidly evolving domain.

2. Literature Review

This literature review distills the latest findings on Ponzi scam detection in Ethereum smart contracts, emphasizing the key vulnerabilities, methods of detection, and the urgent need for formal verification and sound software engineering practices. Susceptibility of intelligent contracts. numerous studies identified serious security vulnerabilities within Ethereum smart contracts that can be used to implement Ponzi schemes[10]. carried out a systematic review of these vulnerabilities and found that most of the fraudulent contracts exploit existing code weaknesses. They noted that it is crucial to know these vulnerabilities to design effective Ponzi scheme detection schemes. In addition, indicated the shortcomings of current smart contract analysis tools and concluded that most are not effective in detecting possible Ponzi scheme-vulnerable vulnerabilities[11]. The conclusions of corroborate this idea, calling for safe development practices to avoid the risks involved in smart contracts. They emphasized a thorough analysis of smart

contract code to detect and fix the vulnerabilities before deployment, important in averting Ponzi schemes. Formal Verification and Detection Methodologies. The need for formal verification when developing smart contracts cannot be stressed enough[12]. noted that techniques for formal verification have the potential to greatly increase smart contract reliability through the identification of vulnerabilities that could be exploited in Ponzi schemes. In like manner, considered the application of formal methods in the form of Isabelle/HOL for Ethereum smart contract byte code verification, an attractive way to strengthen detection mechanisms. Their strategy was to analyze transaction histories and operation codes, which provided a foundation for a methodical detection framework that could be utilized to protect users from abusive behavior. In addition, also suggested a complete strategy for using blockchain data to identify Ponzi schemes, emphasizing the need for ongoing monitoring of smart contracts. Their study recommends a single platform that can act as an early warning system, allowing for the quick detection of possible scams before they grow. The Blockchain Software Engineering Role. The creation of a discipline of Blockchain Software Engineering, as discussed by, is critical for the solution of the distinct problems presented by smart contracts[12], [13]. Analysis of case studies of bugs in smart contract libraries highlights the necessity of following best practices in developing smart contracts to avoid vulnerabilities that could be vulnerable by Ponzi schemes. introduced. Euthanizer, a security auditor that can detect information flow vulnerabilities in smart contracts. This device is especially useful in identifying sophisticated attack vectors for Ponzi schemes since it allows for in-depth blockchain analysis that could reveal potential problems not necessarily obvious at first glance. Gaps in Existing Research Despite the progress in detection methods and the knowledge about vulnerabilities, huge gaps exist in the area of Ponzi scam detection in Ethereum intelligent contracts. Much of the existing research centers on the technicalities of intelligent contracts, vulnerabilities, and detection mechanisms, ignoring the larger picture of user education and regulatory issues. Further empirical research is needed to evaluate the effectiveness of such detection tools and frameworks in actual use. A better understanding of user behavior and the psychological explanations for the success of the Ponzi scam may also help to drive more effective preventative measures.

2.1. Understanding the Ponzi Scam in the Context of Ethereum

Ponzi schemes in the Ethereum community poses a major threat to investors, using the anonymity and decentralized aspect of blockchain technology to spread fraudulent activity. Such schemes, frequently carried out using smart contracts, build an illusion of profitability on the back of funds coming from new entrants to pay out existing investors, thus leading to massive economic losses. Current detection techniques, which primarily scan bytecode and transaction patterns, tend to be missing the interpretability needed for good governance and protection of stakeholders. SourceP introduces a new paradigm by applying the code of intelligent contracts for detection purposes, improving both accessibility and transparency in detecting likely Ponzi schemes[13]. The implications of this study are significant, since the combination of AI-based methods, described in recent reviews on decentralized finance fraud detection, leads towards better approaches to solving such increasing threats[14]. To better visualize this issue, a visual description of the detection process, as described in, effectively sums up the complexity involved in the identification of fraudulent smart contracts.

2.2. Characteristics of Ponzi Schemes and Their Relevance to Smart Contracts

Ponzi schemes are defined by their dependence on the ongoing recruitment of new investors to fund returns to previous ones, presenting a veneer of profitability while eventually failing under the burden of unsustainable financial operations. Decentralized finance faces inherent security risks which resulted in Ponzi scheme development on Ethereum smart contracts [15, 16]. Smart contracts operate independently to conduct transactions so they shield illegal operations that escape regulatory oversight. The authorities now find it more difficult to detect fraudulent operations solely by relying on traditional analysis methods

based on transaction behavior and bytecode analysis. SourceP detection operates as a novel technique to analyze program code and identify Ponzi schemes with enhanced contribution to interpretation while promoting sustainability. SourceP demonstrates its capabilities through 87.2% recall rate performance in experimental findings which presents a crucial solution for protecting investors from new risks. The SourcePs approach becomes more significant due to its visualize compilation of smart contracts and opcode disassembly which explain technical aspects beyond traditional detection solutions [17].

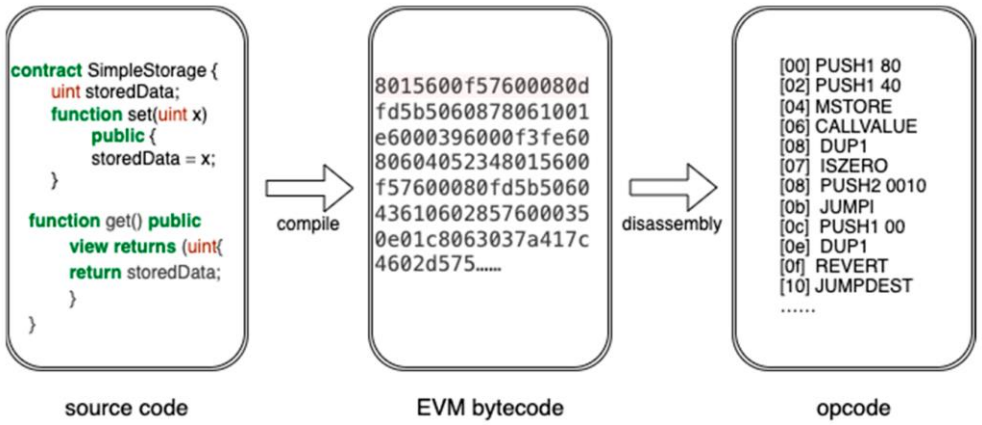

Figure 1: Compilation and disassembly of a Solidity smart contract.

Table 1: Ponzi Scheme Characteristics and Their Relevance to Smart Contracts

Characteristic	Description
Risk-Free Return	Ponzi scams promise unusually high returns with minimal risk, often attracting investors seeking low-risk opportunities. This characteristic is relevant to smart contracts, as they may be used to automate and facilitate such fraudulent schemes, making detection more challenging.
Overly Consistent Returns	Investments in Ponzi scams tend to show consistent positive returns regardless of market conditions, which is uncommon in legitimate investments. Smart contracts can be programmed to display such consistent returns, potentially masking fraudulent activities.
Unregistered Investments	Ponzi scams often involve investments that are not registered with regulatory authorities, lacking transparency and oversight. Smart contracts can facilitate unregistered investments, making it difficult for regulators to monitor and detect fraudulent schemes.

Unlicensed Sellers	Perpetrators of Ponzi scams typically operate without proper licensing, which is a red flag for investors. The pseudonymous nature of smart contracts can enable unlicensed individuals to engage in fraudulent activities without easily identifiable credentials
Hidden and Intricate Strategies	Ponzi scams often involve complex or secretive investment strategies that are difficult for investors to understand, leading to a false sense of security. Smart contracts can be designed with intricate logic, making it challenging for investors to comprehend the underlying mechanisms and assess the legitimacy of the investment.
Difficulty Receiving Payments	Investors in Ponzi scams may experience delays or difficulties in withdrawing their funds, indicating potential fraud. The irreversible and transparent nature of block chain transactions can complicate the process of recovering funds once they have been transferred via smart contracts.

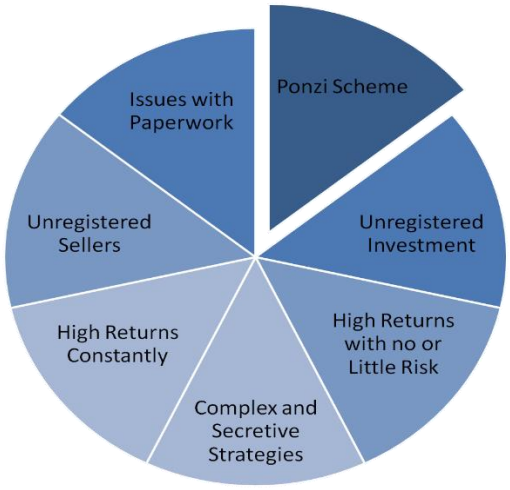

Figure 2: Ponzi Scheme Pie Chart, encompassing the way to recognize patterns

2.3 SourceP: A Tool for Detecting Ponzi Schemes

The development of Source P represents a major breakthrough toward reducing Ponzi scheme spread within Ethereum's extensive network [14]. Source P implements code-based intelligent contract analysis to accomplish new detection standards that surpass traditional bytecode and transaction behavior detection methods thus supporting practical application improvements [15]. The new method allows for both simpler acquisition of information and deeper evaluation of smart contract functional operations. The data flow graph built by the framework provides effective pre-trained processing that leads to outstanding Ponzi scheme identification with F-score 92.4% and recall 90.1%. These solutions bring critical value to manage the billions of dollars vulnerable to fraudulent actions in decentralized finance (DeFi) ecosystems

according to recent research. The presented framework proves SourceP operates as a critical instrument for protecting user financial investments from obscure risks that exist in smart contracts.

3. Methodology

3.1 Identifying Ponzi Schemes

Ponzi schemes on Ethereum typically follow a cyclical investment structure, where new deposits fund withdrawals for earlier participants, eventually collapsing when new investments dry up Some well-known Ethereum Ponzi schemes include:

- Smart Millionaire, which operated under a "double-your-money" scheme.
- For sage, a high-profile smart contract Ponzi that attracted millions in crypto investments.

Key characteristics of Ponzi smart contracts include:

- High referral rewards to incentivize recruitment.
- No external revenue generation apart from new investor deposits.
- Early withdrawal penalties to prevent sudden fund drainage.

3.2 Ponzi Scheme Detection Techniques

Several methodologies have been proposed to identify fraudulent smart contracts:

1. Static Analysis

- **Source code inspection:** Identifying functions that distribute funds recursively.
- **Opcode frequency analysis:** Examining smart contract bytecode patterns.
- **Control-flow graph analysis:** Detecting functions that redirect funds in an unsustainable manner.

2. Dynamic Analysis

- Real-time transaction monitoring to flag anomalies in fund distribution.
- Recursion detection, focusing on contracts that continuously redistribute investments.

3. Machine Learning Approaches

- Graph-based models to classify Ethereum transactions based on known Ponzi patterns.
- Supervised learning trained on labeled datasets of fraudulent and non-fraudulent contracts
- SourceP: A Tool for Smart Contract Analysis

SourceP is an emerging tool that improves Ponzi scheme detection through:

- Code similarity analysis to compare new smart contracts against known Ponzi contracts.
- Pattern-based filtering, focusing on irregular financial flow behaviors.
- Automated reporting, generating risk scores for smart contracts

Studies suggest that Source P outperforms traditional signature-based detection techniques, offering a higher detection rate with fewer false positives. However, challenges remain, including evasion tactics, obfuscated smart contracts, and privacy-enhancing techniques like Zk-SNARKs that hide transaction details.

3.3 Dataflow Chart

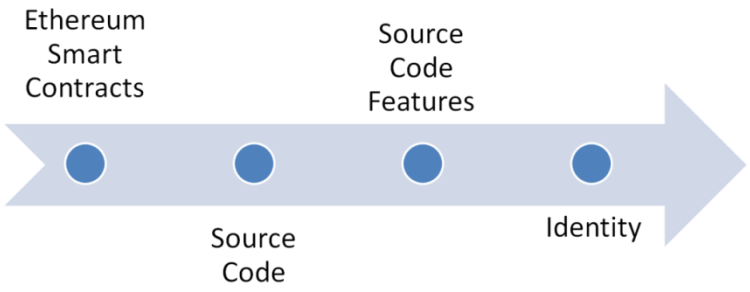

Figure 3Flow chart of the Process

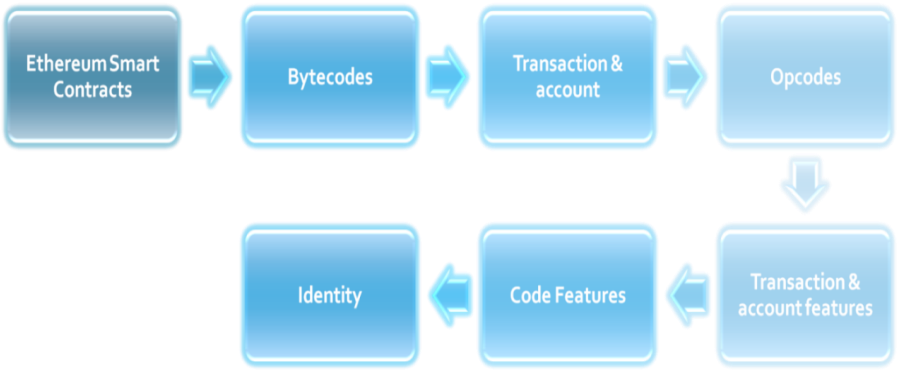

Figure 4: Flowchart of Traditional Method

Data streams, also known as data flow diagrams (DFGs), are graphical representations of dependencies between variables in code. The source of each variable's value is represented by the edge and the nodes that make up this graphic. For code analysis, data diagrams are invaluable tools. Unfortunately, using an abstract syntax tree (AST) data flow diagrams maintains the same structure via various abstract syntaxes of the code. For comprehension of code, this consistent structure provides significant semantic information. Furthermore, data flow diagrams are more efficient in models because they have a simpler structure compared to AST.

3.4 Pre-Trained Model

Pretrained models offer significant advantages for a variety of downstream tasks by leveraging vast parameter storage to fine-tune knowledge for specific applications. The extensive knowledge of the tacit knowledge contained in these parameters was thoroughly verified by empirical analysis. Notable prepared models such as XLNET, Bert, Elmo, and GPT have proven efficacious in many tasks. Several prepared models, including Codebert, Cubert, GPT-C, and Code GPT, are specially designed to display code in machine learning applications. SourceP uses GraphCodeBert, an initial model trained on the CodesearchNet dataset, as an important component of the learning code display. The specific modeling methods used in SourceP are explained below:

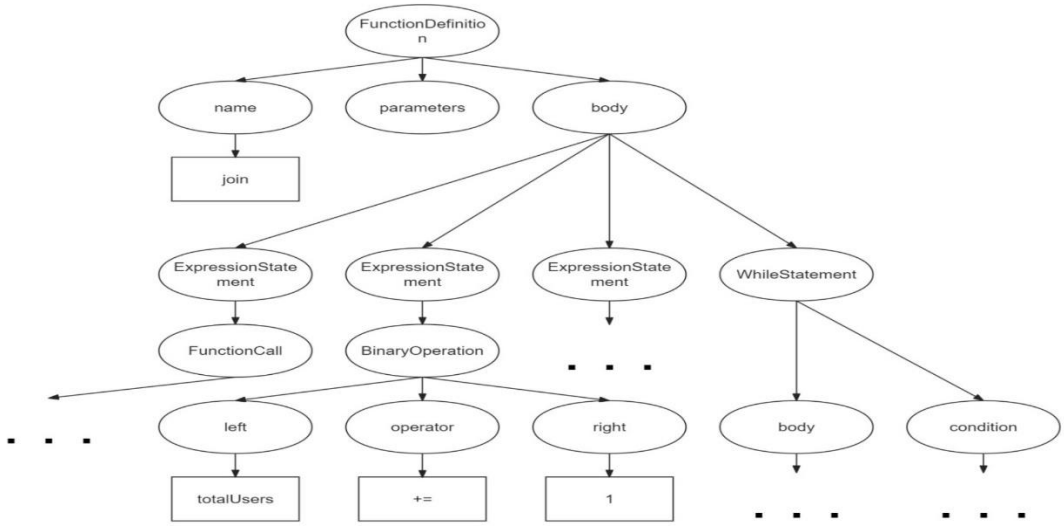

Figure 5: Code Operations Parsing into AST

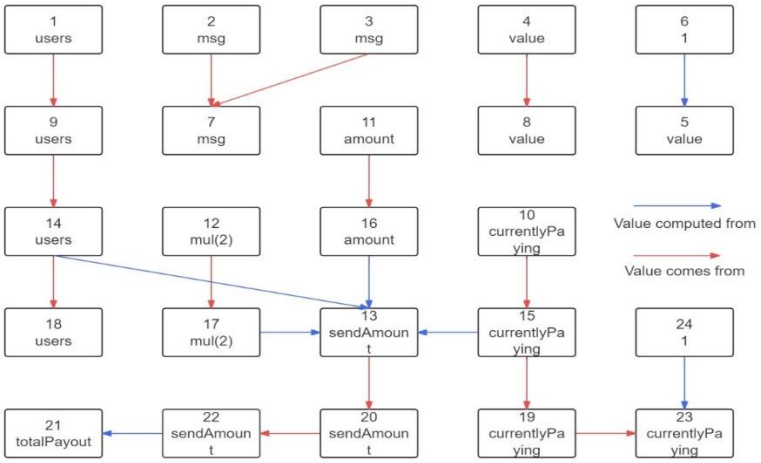

Figure 6: Data Flow Chart of Source Code

3.5 Model Structure

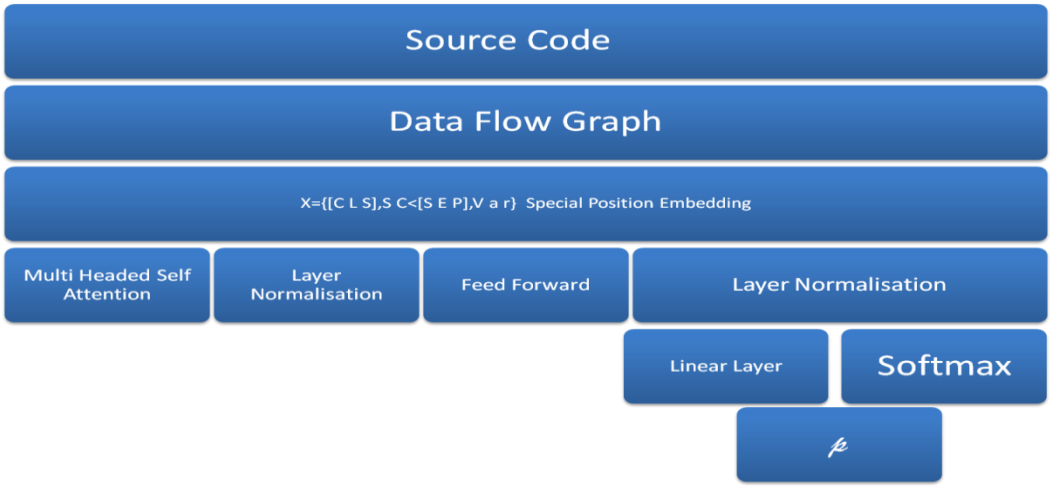

Figure 7: Model Structure of SourceP

Here a comprehensive explanation of SourceP's model structure is provided. A model architecture based on BERT and a multi-layer bidirectional transformer model serves as the foundation for our approach, which is primarily in line with Graph Code Bert. The overall structure is shown in Image 5

3. Result Summary

Using the approach intend to compare the detection performance of SourceP with the performance of existing state-of-the-art methods in this study. Specifically, we rank all contracts based on the block height at the time of smart contract creation. From the first to the 250th position in the training set, smart Ponzi schemes are included, with non-Ponzi smart contracts in between. From position 352 to position 451 of the test set, intelligent Ponzi schemes and the remaining non-Ponzi smart contracts are included. A training set of 6789 intelligent contracts and a test set of 508 smart contracts are produced by this division. This method, when analyzed with a random segmentation, provides a more accurate representation of the model's capacity to identify new Ponzi scam when there is insufficient data on existing schemes. The models we compare include MulCas, SVM-NC, XGBoost-TF-IDF, Ridge-NC, and Sad Ponzi. Ridge-NC and SVM-NC make use of N-gram count features, XGBoost-TF-IDF makes use of TF-IDF features, MulCas uses Developer Feature, and Sad Ponzi finds Ponzi schemes based on smart contract byte code. The comparison results are shown in Table I, and they show that SourceP performs better than any of the other methods in all three metrics. This highlights the effectiveness of Source P in detecting intelligent Ponzi schemes. In all three metrics, the results show that Source P performs better than any other method. Specifically, Source P exhibits a remarkable 21.3% increase in recall and a 12.9% improvement in F-score compared to the current state-of-the-art method, while also enhancing precision. Given the imbalance in the ratio of positive to negative samples, approximately 1:20, it is understandable that the model tends to divide minority prototypes as the predominant one, leading to a remarkable precision score rather than recall.

The model's long-term viability in comparison to other cutting-edge approaches. Model aging is a new problem that has received a lot of attention, even though SourceP has demonstrated exceptional performance in detecting the most recent smart Ponzi schemes. Particularly, there is a significant distinction between the most recent Ponzi schemes within smart contracts and the earlier smart Ponzi schemes. Following (13)method, we carried out an experiment in which we divided the dataset into six parts (P0-P5) based on the block height of the created Ponzi schemes to evaluate SourceP's sustainability. Every 50 smart Ponzi schemes were divided into the dataset, with P0 representing the starting 50 schemes and their non-Ponzi contracts, followed by the rest. Since smart contracts cannot be tampered with, a low block count of creation indicates a quicker creation time. This lets us use previous smart Ponzi schemes to predict new ones and evaluate Source P's viability. Our comparison models, Sad Ponzi and MulCas, were also included in the analysis, with the results presented in Table 2. The increased value of the Ponzi tokens reflects the Ponzi schemes reward as new smart Ponzi schemes based on ERC-20 token trading contracts are implemented. In conclusion, Source P's performance surpasses that of other models, showcasing its capability.

Table 2: State-of-the-art methods were used to compare

Method	Precision	F Score	Recall
MulCas	0.134	0.982	0.270
Ridge-NC	0.62	0.79	0.765
SVM-NC	0.475	0.823	0.733

XGBoost-TF-IDF	0.674	0.951	0.789
Sad Ponzi	0.453	0.829	0.586
Source P	0.9847	0.987	0.981

Table 3: Source P results

Method	Metric	P2	P3	P4	P5
MulCas	Precision	0.87	0.42	0.19	0.23
	Recall	0.67	0.73	0.26	0.19
	F-score	0.34	0.65	0.51	0.21
SourceP	Precision	0.99	0.97	0.85	0.98
	Recall	0.55	0.33	0.93	0.67
	F-score	0.54	0.43	0.85	0.89
Sad Ponzi	Precision	0.33	0.96	0.88	0.97
	Recall	1.00	0.85	0.94	0.85
	F-score	0.5	0.92	0.81	0.90

Table 4: Source P for the three metrics in the dataset that was randomly divided

Method	Recall	Precision	F-score
w/o EdgePred	- 0.867	0.919	0.891
w/o Data Flow	0.821	0.914	0.860
Source P	0.887	0.956	0.918
-w/o Node Align	0.806	0.909	0.847

TABLE III: Ablation experiments without pre-training tasks or data flow compared to the three metrics

Method	F-score	Recall	Precision
Source P	0.91	0.92	0.924

TABLE 4 Here ablation experiment is conducted to determine how pre-training tasks and data flow affected smart Ponzi scam detection performance. this was accomplished by eliminating two data flows and pre-training tasks separately. Performance loss occurred in the model according to Table III since data flow absence and pre-training elimination impair model effectiveness in different ways. The model benefits from these two components which play essential roles in achieving better performance outcomes.

4.1 Functionality and Effectiveness of Source P in Analyzing Smart Contracts

The ability of Source P to examine smart contracts stands as the main component in resolving vulnerabilities that exist within Ethereum's volatile environment through Ponzi scheme elimination. To enhance its suspicious contract detection process, Source P uses complex analysis techniques that combine machine learning algorithms with op code context assessment. The n-gram algorithm plays a crucial role

because it enables a detailed examination of contract op codes to identify malicious patterns. The implementation of adaptive synthetic sampling techniques addresses class imbalance challenges to raise both the precision and reliability of the model. The integration of two detection approaches helps identify suspicious contracts while fixing the existing detection system's deficiencies regarding feature selection optimization. Therefore, Source P emerges as an effective tool in the ongoing fight against deceptive smart contracts, aligning with contemporary needs for robust security measures.

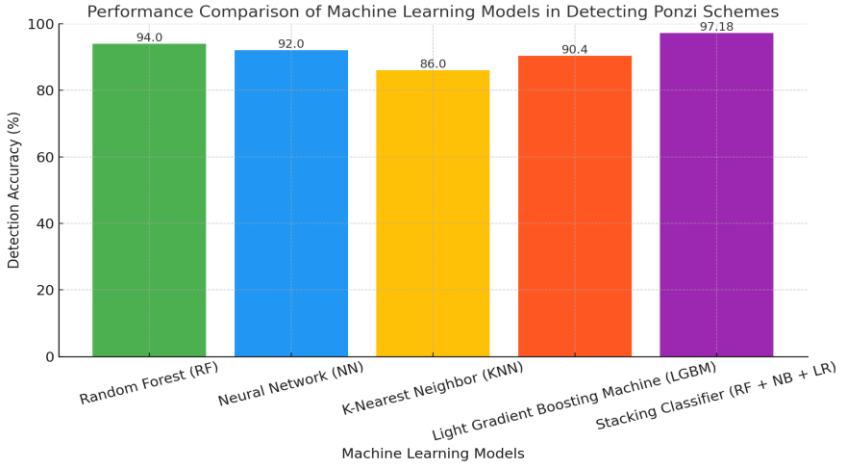

Figure 8Machine learning models results

Figure 8 shows a comparison of the detection performance of several machine learning models for finding Ponzi scams in Ethereum smart contracts. It illustrates that the highest accuracy was delivered by the Stacking Classifier at 97.18%, followed by the Random Forest at 94%. Neural Network and Light Gradient Boosting Machine also demonstrated good performances, whereas K-Nearest Neighbor was less effective at 86%. The visualization conveys the success of sophisticated methodologies in improving detection accuracy.

4.2 Vulnerability Analysis

The proliferation of intelligent contracts on blockchain platforms like Ethereum has ushered in a new era of decentralized applications (dApps), yet it has also opened doors to novel security vulnerabilities, particularly those exploited in Ponzi schemes Detecting such schemes necessitates a multi-faceted approach, focusing on identifying Source P vulnerabilities that can be leveraged for fraudulent activities. The early existence of tools and programming languages for smart contract development produces complex difficulties that can cause users to misunderstand standard features. A strong approach to manage these weaknesses needs to incorporate both manual code analysis along with automated diagnosis systems along with auditing programs. Professional code reviews perform by experienced developers constitute the essential method to discover possible issues within smart contract programs. The vulnerability detection capabilities of MythX along with Slither as well as Remix are significantly improved by automated analysis methods and through static and dynamic analysis. Static code analysis conducts examination of program code while it remains inactive to discover security holes through the identification of predefined coding styles and weaknesses. Running smart contracts within a controlled testing environment during dynamic analysis enables the identification of vulnerabilities that appear while the code executes. Mathematical approaches for proving smart contract correctness through formal verification provide defense against programming vulnerabilities on a high level.

4.3 Limitations of Detection of Ponzi Schemes

Ethereum serves as a current infrastructure for developing decentralized applications together with intelligent contracts. Multiple entities have become interested in Ethereum's inventive technology because it brings restructuring opportunities for nefarious ends. Users on the Ethereum network encounter a significant risk from Ponzi schemes that are integrated within smart contracts. These criminal activities create false investment promises before fresh investments vanish into thin air. The development of Source P limitations represents a research tool designed to detect Ponzi schemes by analyzing intelligent contract identity codes. Developers can protect themselves and their clients against Ponzi scheme frauds when they learn to identify signature features that define such frauds. All stakeholders of Ethereum must actively monitor unidentified threats because their commitment ensures sustainable platform growth together with success. The structural limitations of Ethereum smart contracts in handling fraudulent activities stem from their fixed nature along with the lack of centralized authority. Once deployed on a blockchain network an intelligent contract continues without any ability to modify or terminate it thus making unauthorized intervention in fraud cases extremely difficult. Some complex Ethereum smart contracts display hidden characteristics that obscure investment terms and risks from investors which increases their susceptibility to Ponzi schemes. Investors need to base their Ethereum smart contract decision-making on their own diligence because code review and audit activities help scout potential scams but cannot replace investor focus. Ongoing research with regulatory oversight is necessary to protect investors from Ponzi schemes in the fast-evolving digital world because Ethereum smart contracts currently do not provide sufficient prevention.

5. Conclusion

The advancement of Ethereum into decentralized finance demonstrates that effective tools to detect Ponzi scams in its intelligent contracts are essential right now. Source P demonstrates superior performance metrics over traditional detection approaches through its combination of source code analysis and pre-training models which simultaneously generates high interpretability of fraud detection with a detection efficiency reaching 90.2% recall and 92.7% F-score. The shifting DeFi industry demands new approaches to address its detection needs as detailed in a formal classification of DeFi project fraud types. The research results explain the Source P model structure while confirming AI applications for fraud identification systems to show how digital technology brings a revolutionary standard of asset protection against unlawful exploitation. The discovery of Ponzi schemes in Ethereum smart contracts creates critical issues about how the platform will develop about user confidence and sustainable operations. Source P emerges at a time when financial fraud through smart contracts rises thus establishing advanced security measures which address this urgent need. The combination of source code analysis and innovative pre-training models in Source P offers improved interpretability along with sustainability over conventional byte code and transaction-based detection methods with their unclear context. Enhanced detection through this solution effectively defends investors from substantial financial damage as it strengthens the entire Ethereum platform. This analysis proves that strong detection and response methods for Ponzi schemes will build a more extensive infrastructure for decentralized finance.

References

[1] Y. Zou, “Detecting Vulnerabilities In Ethereum Smart Contracts Through Execution Trace Analysis,” University of Guelph, 2024.

[2] A. Alghuried, “Learning-Based Ethereum Phishing Detection: Evaluation, Robustness, and Improvement,” 2024.

[3] J. Liu, J. Chen, J. Wu, Z. Wu, J. Fang, and Z. Zheng, “Fishing for fraudsters: Uncovering Ethereum phishing gangs with blockchain data,” IEEE Transactions on Information Forensics and Security, vol. 19, pp. 3038–3050, 2024.

[4] C. Wu et al., “Token scout: Early detection of Ethereum scam tokens via temporal graph learning,” in Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security, 2024, pp. 956–970.

[5] B. Weng and J. Ma, “Semantic-Based Detection of Ponzi Smart Contracts: Enhancing Bytecode Analysis with Source Code,” in 2024 4th International Conference on Electronic Information Engineering and Computer (EIECT), 2024, pp. 326–331.

[6] L. Wang, H. Cheng, Z. Sun, A. Tian, and Z. Yang, “PSPL: A Ponzi scheme smart contracts detection approach via compressed sensing oversampling-based peephole LSTM,” Future Generation Computer Systems, vol. 166, p. 107655, 2025.

[7] Y. Qu, X. Si, H. Kang, and H. Zhou, “Detecting Ethereum Ponzi Scheme Based on Hybrid Sampling for Smart Contract.,” Computers, Materials & Continua, vol. 82, no. 2, 2025.

[8] M. Fawad, M. Asfandyar, Z. Ullah, A. Ullah, M. N. Ullah, and others, “Finding Influential Nodes in Ethereum Using Machine Learning,” Spectrum of engineering sciences, vol. 3, no. 1, pp. 402–424, 2025.

[9] M. T. Tran, N. Sohrabi, Z. Tari, Q. Wang, and X. Xia, “Slow is Fast! Dissecting Ethereum’s Slow Liquidity Drain,” arXiv preprint arXiv:2503.04850, 2025.

[10] A. Kumar, A. Paliwal, B. Tanwar, G. Maheshwari, and S. Maheshwari, “Harnessing Blockchain and Smart Contracts for Next-Generation Digital Identity: Enhancing Security and Privacy”.

[11] B. Alotaibi, “Cybersecurity Attacks and Detection Methods in Web 3.0 Technology: A Review,” Sensors, vol. 25, no. 2, p. 342, 2025.

[12] B. C. Das et al., “Detecting Cryptocurrency Scams in the USA: A Machine Learning-Based Analysis of Scam Patterns and Behaviors,” Journal of Eco humanism, vol. 4, no. 2, pp. 2091–2111, 2025.

[13] M. Bresil, P. Prasad, M. S. Sayeed, and U. A. Bukar, “Deep Learning-based Vulnerability Detection Solutions in Smart Contracts: A Comparative and Meta-Analysis of Existing Approaches,” IEEE Access, 2025.

[14] A. Jyoti, P. Gupta, S. Gupta, H. Khatter, and A. Mishra, "Inherent Insights using Systematic Analytics of Developments Tools in Ethereum Blockchain Smart Contract," Recent Advances in Electrical & Electronic Engineering, vol. 18, no. 2, pp. 135–146, 2025.

[15] G. Chaudhari, "Trustless Contracts for AI Model Exchange in Banking: Secure Model Evaluation and Monetization on the Ethereum Blockchain," Authorea Preprints, 2025.

[16] W. Khiari, A. Lajmi, A. Neffati, and A. El Fahem, "Cryptocurrency fraud and its effects on price volatility in the cryptocurrency market," Journal of Chinese Economic and Foreign Trade Studies, 2025.

[17] S. Chen and F. Li, "Ponzi scheme detection in smart contracts using the integration of deep learning and formal verification," IET Blockchain, vol. 4, no. 2, pp. 185–196, 2024.