BLOCKCHAIN AND ARTIFICIAL INTELLIGENCE FOR INTELLIGENT AND SECURE FINANCIAL TRANSACTIONS: AN IMPLEMENTATION PERSPECTIVE WITH PYTHON

Vasil Spasov[1],

e-mail: vasil.spasov@*unwe.bg*[1]

Абстракт

Традиционните финансови системи страдат от бавни междубанкови преводи, високи такси и рискове от измами, като централизираните им архитектури създават критични уязвимости. Настоящото изследване предлага архитектура за децентрализирана, интелигентна система за финансови транзакции, базирана на блокчейн и изкуствен интелект, като илюстрира практическата ѝ реализация чрез Python. Целта е система със значително намаляване на транзакционните разходи и автоматизирана детекция на измами. Python е подходящ за тази цел, благодарение на богатата си екосистема от библиотеки.

Abstract

Traditional financial systems suffer from slow interbank transfers, high fees, and fraud risks, with their centralized architectures creating critical vulnerabilities. This study proposes an architecture for a decentralized, intelligent financial transaction system based on blockchain and artificial intelligence, demonstrating its practical implementation using Python. The goal is a system that significantly reduces transaction costs and enables automated fraud detection. Python is suitable for this purpose due to its rich ecosystem of libraries.

Key words: Blockchain, Artificial Intelligence, Python, Financial Transactions, Machine Learning

JEL: G21, C88, O33

1. Introduction

Contemporary financial systems rely on centralized intermediary institutions, resulting in slow and expensive transactions, as well as increased vulnerability to cyber attacks. Traditional methods for combating fraud are no longer adequate against modern threats.

Blockchain technology offers a decentralized and transparent alternative approach, eliminating intermediaries through cryptographically secure mechanisms. Simultaneously, artificial intelligence provides capabilities for big data analysis and real-time risk prediction, complementing blockchain solutions.

Python emerges as a suitable tool for implementing this convergence with its rich ecosystem of libraries: scikit-learn for machine learning, web3.py for Web3 development, Vyper for smart contracts. This report illustrates why Python is an excellent choice for creating hybrid blockchain-AI systems

Research Objective:

To propose a system architecture that combines blockchain and artificial intelligence for secure financial transactions and to demonstrate its practical implementation using Python.

[1] PhD, Faculty of Finance, University of National and World Economy

This research aims to design and develop an integrated system architecture that effectively combines blockchain technology with artificial intelligence to create a secure, intelligent, and efficient framework for financial transactions. The solution that harnesses the complementary strengths of both technologies—leveraging blockchain's inherent advantages of decentralization, immutability, and cryptographic security while integrating AI's advanced capabilities in machine learning, predictive analytics, and adaptive pattern recognition. The primary focus is on building a functional system that addresses critical challenges in modern financial operations, including real-time fraud detection. The goal is to provide a scalable, adaptable model that bridges theoretical innovation with practical application, offering significant improvements in financial security, cost reduction, and system intelligence while maintaining the flexibility to evolve with emerging technological standards and regulatory requirements.

Methodology:

The research employs an analytical approach for designing a system architecture that integrates blockchain networks (Ethereum), smart contracts (Vyper), scikit-learn machine learning algorithms, and decentralized storage (IPFS). The methodology includes a review of relevant literature, conceptual design, and evaluation of practical applicability through Python.

2. Literature Review

Blockchain technology, introduced with the Bitcoin protocol by Nakamoto (2008), represents a decentralized ledger with fundamental characteristics: immutability, transparency, and absence of centralized control. The Ethereum platform by Buterin (2014) expands this concept through smart contracts - self-executing programs that automate contractual agreements without intermediaries.

In the field of artificial intelligence for financial security, West and Bhattacharya (2016) provide a comprehensive review of intelligent fraud detection methods, demonstrating the superiority of algorithms over traditional systems. Modern machine learning algorithms recognize fraudulent transactions with high accuracy.

Liang et al. (2025) extend the research in blockchain context, showing how machine learning detects Ponzi schemes in Ethereum through analysis of behavioral patterns of smart contracts and transactions.

The synergy between blockchain and AI has been analyzed by Salah et al. (2019), who identify how blockchain addresses AI challenges related to model transparency and data trust, while AI enhances the efficiency of blockchain systems through intelligent optimization of consensus mechanisms and automated anomaly detection.

From an implementation perspective, Python dominates machine learning, with Pedregosa et al. (2011) documenting scikit-learn as a powerful library providing a rich set of algorithms. This Python ecosystem extends to blockchain technologies through the Vyper language, which offers Python-like syntax for smart contract development with focus on security and code auditability.

Wang et al. (2021) examine the NFT potential for digitization of unique assets and documents in the financial sector, including securities, contracts, and proof of ownership.

Scalability challenges have been analyzed by Croman et al. (2016), who identify fundamental limitations of blockchain networks in processing high volumes of transactions and propose second-layer solutions. Regulatory aspects are the subject of Finck's (2019) research regarding the tension between blockchain immutability and GDPR requirements, proposing approaches for reconciliation through techniques such as cryptographic erasure and off-chain storage of personal data.

3. Python-Based Architecture for Blockchain-AI Financial Systems

The proposed architecture integrates blockchain and artificial intelligence using Python as a central tool. The system combines decentralized governance with intelligent automation and multi-layered security.

The blockchain layer is implemented through the web3.py library, which serves as a standard interface for Python applications to interact with Ethereum and compatible blockchain networks. The library provides complete functionality not only for executing transactions and working with smart contracts, but also for real-time event monitoring. For smart contract development, Vyper is used - a Python-like programming language specifically designed for writing secure blockchain applications, which eliminates complex and risky constructs, making smart contracts safer and easier to create.

The machine learning module analyzes transactions in real-time to detect fraud. To assess risk, the module analyzes historical data and multiple parameters such as transaction value and frequency, geographic location, and user behavioral patterns.

For storing large volumes of data that cannot be efficiently recorded directly on the blockchain due to high costs and memory limitations, the system integrates IPFS (InterPlanetary File System). This way, only encrypted data hashes are recorded on the blockchain, while the actual files are stored in the decentralized IPFS network. This hybrid approach ensures data security and immutability while maintaining efficiency and low operational costs.

Before any financial transaction is finally executed, it undergoes an automated risk assessment system based on artificial intelligence. This validation module uses a pre-trained machine learning model created and trained on extensive historical data from past transactions, including both legitimate and fraudulent operations. The model continuously learns from new data to improve its accuracy and adapt to new fraud patterns.

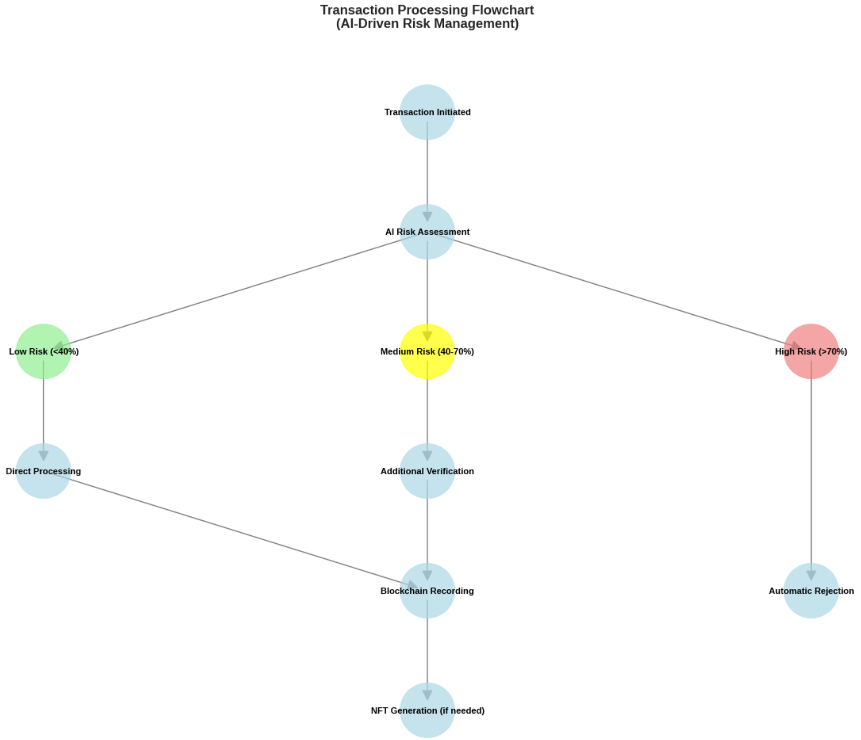
The system analyzes multiple parameters in real-time to generate a probabilistic risk assessment for fraud. Among the analyzed factors are: the transaction amount and its deviation from the user's typical values; the frequency of transactions and their temporal patterns; the geographic location of the operation and its compatibility with previous locations; the user's behavioral patterns and their consistency; the device type and IP address used for the transaction; as well as historical data about counterparties and their risk profiles.

For final decision-making, the system uses a three-tier threshold system that balances security and user convenience. For transactions with extremely high risk (over 70% probability of fraud), the system automatically blocks the operation without human intervention. For medium-risk transactions (between 40% and 70%), the system flags them for additional verification by security specialists or requires additional identity authentication methods. Low-risk transactions (below 40%) are processed directly and immediately without additional delays.

After a transaction successfully passes all checks and is executed, it is recorded on the blockchain as a permanent and immutable record. Each record contains a unique cryptographic hash that serves as a digital fingerprint of the transaction. This hash is generated through cryptographic algorithms and guarantees that even a minimal change in the transaction data will result in a completely different hash. All network participants can verify and authenticate the transaction through this hash, ensuring complete transparency and traceability.

In specific cases where the transaction involves ownership transfer, fulfillment of contractual conditions, or creation of a digital asset, the system automatically generates a Non-Fungible Token (NFT) through a specialized ERC-721 smart contract. This NFT serves as digital proof of ownership or contract fulfillment. The token contains metadata describing the specific asset or condition and is stored in the owner's wallet. The metadata may include a detailed description of the asset, deal terms, creation timestamp, as well as history of ownership transfers. This mechanism provides secure, transparent, and verifiable proof of rights and obligations arising from the financial transaction.

The system implements a continuous learning mechanism for the AI model, where new data from transactions verified as legitimate or fraudulent is periodically collected, and the model is retrained to adapt to evolving fraud patterns. This process can be automated when a certain volume of data is reached.



Source: Author's development

Figure 1: Transaction Processing Flowchart. AI Risk Management

4. Conclusion

The study demonstrates the implementation of a hybrid system for secure financial transactions using Python, integrating blockchain and artificial intelligence. Tools such as web3.py, Vyper, and scikit-learn were employed, providing a comprehensive solution with fundamental advantages: reduced costs and processing time, automated fraud detection, and complete transaction traceability.

The selection of Python for implementing an intelligent financial transaction system is not only technologically feasible but also strategically justified, opening opportunities for innovation, broad accessibility, and easy integration across multiple domains. The combination of blockchain's immutability and transparency with AI's intelligent analytical power, implemented through Python's versatile ecosystem, represents a genuine paradigm shift toward a more efficient financial system.

References

1. Nakamoto, S. (2008). "Bitcoin: A Peer-to-Peer Electronic Cash System." Bitcoin.org.
2. Buterin, V. (2014). "A Next-Generation Smart Contract and Decentralized Application Platform." Ethereum White Paper.
3. West, J., & Bhattacharya, M. (2016). "Intelligent Financial Fraud Detection: A Comprehensive Review." Computers & Security, 57, 47-66.
4. Liang, R., et al. "(2025). Towards Effective Detection of Ponzi Schemes on Ethereum with Contract Runtime Behavior Graph. ACM Transactions on Software Engineering and Methodology, 34(4), 106, 1-32. https://doi.org/10.1145/3707458
5. Salah, K., et al. (2019). "Blockchain for AI: Review and Open Research Challenges." IEEE Access, 7, 10127-10149.
6. Croman, K., et al. (2016). "On Scaling Decentralized Blockchains." Financial Cryptography and Data Security, 106-125.

7. Finck, M. (2019). "Blockchain and the General Data Protection Regulation." European Parliamentary Research Service.Scientific Foresight Unit (STOA) PE 634.445 – July 2019
8. Wang, Q., et al. (2021). "Non-Fungible Token (NFT): Overview, Evaluation, Opportunities and Challenges." arXiv preprint arXiv:2105.07447.
9. Web3.py Documentation. https://web3py.readthedocs.io
10. Vyper Documentation. https://vyper.readthedocs.io
11. Pedregosa, F., et al. (2011). "Scikit-learn: Machine Learning in Python." JMLR, Journal of Machine Learning Research, 12, 2825-2830.