Designing an AI Neural Network–Enhanced Blockchain for Highly Secure Transactions and Trusted Business Communication

Krisztián Bálint
Keleti Károly Institute of Business Sciences and Digital Knowledge, Obuda University,
Tavaszmező u. 15-17, H-1084
Budapest, Hungary
balint.krisztian1@uni-obuda.hu

Fruzsina Nagy
Keleti Károly Institute of Business Sciences and Digital Knowledge, Obuda University,
Tavaszmező u. 15-17, H-1084
Budapest, Hungary
nagy.fruzsina@kgk.uni-obuda.hu

***Abstract*—Cryptographic solutions used in blockchains provide robust protection against tampering and several classes of cyberattacks, however, they do not inherently prevent fraud or abuse. This study proposes a blockchain architecture that can monitor transaction processes using an embedded neural AI component and detect, in real time, transactions that indicate fraudulent activity. In the proposed KGK AI Transactional Blockchain, a neural network complements the blockchain's guarantees of data integrity and authenticity, but it cannot distinguish between transactions that are formally valid yet malicious or anomalous. An autoencoder-based AI layer learns normal transaction patterns and can provide early warnings when deviations appear that may indicate fraud, compromised keys, or attack behavior. The AI layer issues an approval (i.e., a risk-based sign-off) only if the transaction matches "normal behavior" and is not considered commercially risky.**

Keywords—artificial intelligence, blockchain, neural network, smart contract risk assessment

I. Introduction

Information security is a critical component of enterprise-level information systems, due to the growing number of data losses, cyberattacks, and security incidents [1]. The integration of blockchain and artificial intelligence (AI) has attracted significant research attention in recent years, because such integration may improve the security, efficiency, and productivity of applications in business environments characterised by volatility, uncertainty, complexity, and ambiguity [2]. In blockchain systems, data is stored in a decentralized manner, but there is still a risk of compromise, so the data must be protected appropriately. One possible solution is pseudonymization to protect sensitive data without compromising its usefulness and restricting its access [3]. The prominent purpose of using blockchain technology for maintaining accounting records is to create and establish trust and a trust network either with or without a trusted party involved [4]. Smart contracts, as self-executing agreements on blockchain platforms, improve transaction efficiency by bypassing intermediaries [5]. However, smart contracts can also be compromised or attacked if the contract algorithm is flawed. Smart contracts are characterized by the fact that their contents are executed automatically. These solutions revolutionized decentralized finance by enabling trustless transactions and the operation of decentralized applications. Despite their advantages, the security of smart contracts remains a concern, leaving them vulnerable to malicious actors [6]. For the sake of blockchain and smart contracts, it is necessary to explore neural networks and AI-based options to make decentralized data storage even more secure, which could increase trust in blockchain-based business transactions. Blockchain solutions allow you to send and receive money, which is one of the cornerstones of transactional blockchains. As more businesses go online, fraud and anomalies in online systems are also on the rise. Fraud detection systems that rely on static rules created by human experts have been used to combat online fraud. Nevertheless, organizations face many fraudulent activities in online transactions that need to be minimized [7]. Cybersecurity experts have realized that with increased use of cloud computing comes a greater need for IDS (Intrusion Detection Systems) to identify any possible breaches and ensure safety. While there were several security measures in place, none could identify threats amidst dense clouds and network traffic, but these limitations are addressed by the development of AI-based IDS for quick and effective analysis of attacks. [8]. In the Ethereum Solidity language, both call and delegatecall are low-level functions used to facilitate interaction between smart contracts. The primary distinction between them is that delegatecall only utilises the code from the target address, but the execution context remains within the caller contract. Consequently, delegatecall is frequently employed in scenarios where multiple contracts invoke a library contract, thereby avoiding redundant deployment and saving resources consumed during the deployment process. However, delegatecall is a dangerous function; by passing in specific parameters, the attacker can potentially manipulate the current contract state (such as balance or storage variables). [9]. For these reasons, it is important to create a blockchain that employs forward-looking security solutions that use added data protection solutions with the support of artificial intelligence. The convergence of blockchain technology and artificial intelligence is having a transformative impact, opening a new era of security and transparency in the financial sector [10]. The research questions that guided this study were as follows:

Research Question: Can a multi-layer AI anomaly detection framework integrate into a permission blockchain to improve

the detection of anomalous transactions while maintaining acceptable blockchain performance?

The research is structured organized around the following components:

- Designing an AI neural network-based blockchain,
- Contractual risk assessment in KGK AI transactional blockchain
- Artificial intelligence powered protection,
- KGK Supply Blockchain in practice,
- Transactional blockchain trusted business communication in the KGK AI transaction blockchain.

The research will create a KGK AI Transaction Blockchain, which will be tasked with securely storing transaction data as a first step, and then expanding the blockchain with the support of artificial intelligence and connecting a smart contract, the combined goal of which is to filter out suspicious transactions.

II. Designing an AI Neural Network Based Blockchain

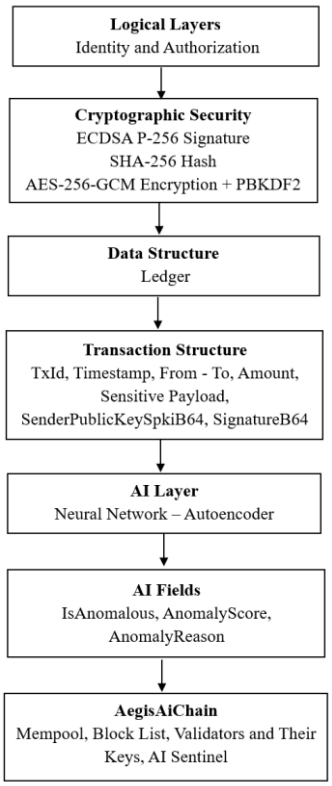
When creating a transactional blockchain, an important consideration must be to establish a tamper-proof ledger that can execute transactions quickly and efficiently. The essential to ensure that only known participants can write to the ledger, especially in the role of validators and that write permissions are strictly controlled. This can be achieved in the following case:

- Permissioned membership, this is the validator allowlist: The Validator(Id, PublicKey) list records who are validators. Only those who are on the allowlist can produce a block.
- Designated proposer, also known as round-robin: ExpectedProposerForNextBlock() tells who is next. If he is not the one trying to produce a block, the system will reject it.
- Transaction-level identity, such as ECDSA: Every transaction must have a SenderPublicKeySpkiB64 and a SignatureB64.

In addition, proper handling of confidential data is also important in a transactional blockchain, so designing the layers appropriately is crucial. In the cryptographic layer, using ECDSA on P-256 (secp256r1 or prime256v1), an elliptic-curve digital signature scheme can be implemented, where the private key is used to sign, and the public key is used for verification. In the KGK AI Transactional Blockchain, AES-256-GCM (Galois Counter Mode) and PBKDF2 (Password-Based Key Derivation Function 2) were used in tandem, as their combined use is suitable for the encrypted storage of sensitive data. By using the AES 256-bit key in GCM mode, the KGK AI blockchain provides both encryption and tamper protection. Finally, the use of PBKDF2 in the blockchain can generate a strong cryptographic key from a weak human password, because the salt is generated during encryption, which is added to the ciphertext. This solution prevents rainbow table attacks and ensures that the same password always gives a different key. During the iteration number, PBKDF2 repeats the hash-based operation many times, which slows down the key derivation, which makes it more difficult for the attacker to decipher the password.

So far, the most widely adopted approach has been the so-called “classical” blockchain security model, which relies on hashing, digital signatures, and a PoA consensus mechanism to protect cryptographic integrity. However, it does not automatically safeguard the system against fraud, compromised keys, abuse, or behavior-based attacks. The KGK AI blockchain’s neural AI solution adds an extra security layer to address this issue, as it can detect post-hoc data tampering, supervise the authentication process, and control the block creation workflow. The following Fig. 1 shows the structure of the KGK AI Transactional Blockchain.

Fig. 1. The construction and structure of the AI blockchain



III. Contractual Risk Assessment in KGK AI Transactional Blockchain

Contract Risk Assessment acts as pre-execution security gate in the smart contract execution process. Before the contract is executed, determines the risk profile of the given operation and accordingly assigns dynamic execution conditions to it. The KGK AI Blockchain system evaluates before execution what business, technical, and security risks a transaction (in this case, a smart contract operation) may carry, and tightens the execution conditions accordingly, such as requiring multi-signature approval, applying a time lock, enforcing manual approval, and setting limits. The purpose of this blockchain solution is not to “ban” contracts, but to complement the blockchain’s cryptographic integrity, such as hashing, digital signatures, and the PoA consensus mechanism, with behavior and content-based decision-making, so that operations that are formally valid but

business-wise or security-wise risky are allowed only under stricter controls.

In this KGK AI Blockchain system, the smart contract operation is the transaction call, which includes the following elements, such as:

Fig. 2 below shows the Smart Contract operations within the KGK AI blockchain.

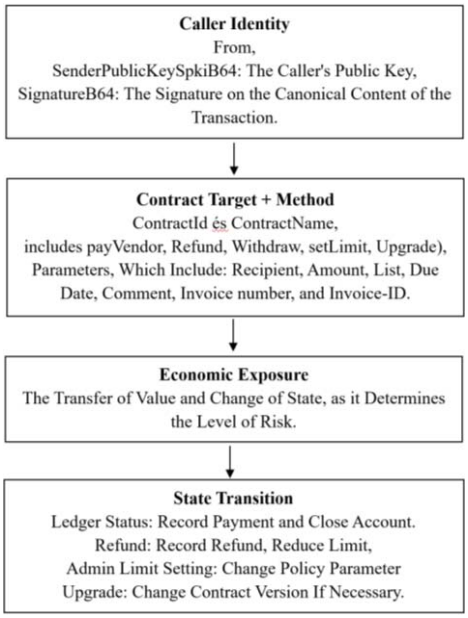


Fig. 2. Smart Contract operations in the KGK AI blockchain

The KGI AI Transaction blockchain smart contract contains two types of features. These are as follows:

- The features of the smart contract call, which include what fields or characteristics describe the operation, such as ContractId, Method, parameters, and
- AI risk assessment features that incorporate numerical, model-friendly features that RiskEngine and AiSentinel calculate from the call.

The smart contract call features are an important part of party identification and contract version recording. This is especially important after an upgrade, as contracts are frequently modified to meet demand. However, when changing versions, the risk increases, as new code appears, and therefore new rules. However, these are necessary, as economic parameters are constantly changing, such as limits and fees.

Risk assessment features monitor speed and time window activities.

- The fromOutVelocity_60s monitors the number of outgoing transactions of the caller in the last 60 seconds, as so-called rapid sweeps are common with compromised keys.
- The toInVelocity_60s monitors the number of incoming transactions for the recipient in the last 60 seconds, as in fraud cases, many incoming transactions often go to so-called aggregator addresses.
- VelocityRatio monitors the relative jump, which can be more effective in indicating an anomaly than the absolute number.

IV. Artificial Intelligence-Posered Protection

KGK's AI blockchain-powered protection solution is a neural autoencoder-based anomaly detection layer that calculates a risk score based on features derived from transaction behavior, and then implements risk-based enforcement, such as timelock, multi-sig, manual review, and quarantine solutions, in conjunction with the policy engine.

An autoencoder is a neural network that learns the pattern of "normal" transactions by trying to reconstruct the input back to itself. In the case of a so-called normal transaction, the reconstruction error is small, while in the case of an unusual transaction, this means a high reconstruction error. The KGK AI blockchain uses an unsupervised solution, which means that there is no need for labeled fraud data, which is often missing in real systems, but only learns from normal operation. Unsupervised machine learning is a learning method where there are no predefined labels or outputs. The model looks for patterns, structure, groups, and differences in the data. The applied artificial intelligence can cluster, as it examines which records are like each other. It looks for anomalies, thereby continuously examining what deviates from normal behavior? Its goal is to identify rare, different, suspicious patterns. The following anomaly detection methods are available on the blockchain:

- Isolation Forest (iForest),
- One-Class SVM,
- Autoencoder (neural network).

Isolation Forest (iForest) is an unsupervised anomaly detection method that models normal transaction and operational behavior rather than learning explicit "fraud" labels. Because anomalies are typically rare and differ from most observations, they are easier to isolate in feature space. iForest builds an ensemble of isolation trees using random features and split selection; anomalous points usually require fewer splits to isolate, resulting in shorter average path lengths. In blockchain systems, iForest can function as an AI-based risk gate by producing a risk score before executing a smart contract call or committing a transaction to a block.

In the KGK AI Transaction Blockchain, iForest performs scoring. It monitors how many steps or splits a transaction can be isolated in trees built with many random splits. The fewer splits needed on average, the more suspicious the given transaction. It traverses each isolation tree from root to leaf based on the transaction features and then counts how many branches it has taken. The resulting value is the path length h(x). It is calculated using the following formula, where:

- x: The record under investigation, which is a feature vector of a transaction,
- h(x): The path length in an isolation tree indicates how many branches, or splits, x goes through before it reaches a leaf.

- E[h(x)]: Average of path lengths across all trees.
- c(n): A normalization constant that approximates the average path length in points when a tree is built from n samples.

$$s(x)=2^{-\frac{E[h(x)]}{c(n)}} \quad (1)$$

The blockchain interprets the following s(x) results as follows:

- s(x) close to 0: The point is difficult to isolate because many splits are required. This result is typically in the dense, normal region, which represents low risk. The transaction characteristics fit the learned normal patterns.
- s(x) ~ 0.4 – 0.6: Borderline or average isolability. This is the average value in many implementations. Not an outlier. It means a moderately secure transaction. A new type of legitimate behavior may appear on the blockchain. This could be a new partner, a new contract method, or a one-time larger amount.
- s(x) ~ 0.60 – 0.80: The point is very easy to isolate, as there are few splits. It assumes a high risk, indicating an anomaly. The transaction differs in several dimensions or shows a rare combination. If such a result appears after creating a new blockchain, it is likely because there have been few similar cases in the data set so far.

One-Class SVM (OC-SVM) is an unsupervised or semi-supervised anomaly detection method that is typically used to learn only from legitimate data. The goal of OC-SVM is to learn the range of normal behavior and label anything that deviates from it as an anomaly. It uses the following decision function, where:

- g(x): Decision function value,
- f(x) = It represents a label and class that is used to recognize the anomaly.
- Sign is a mathematical function with a sign that only tells you whether a number is positive, negative, or zero.

$$f(x)=\text{sign}(g(x)) \quad (2)$$

Using the above formula, OC-SVM places most of the normal data into a closed region and marks points outside the region as anomalies.

In the KGK AI Transaction Blockchain, it is advisable to use Isolation Forest and the One-Class SVM solution simultaneously because they monitor the same transaction differently. By using them together:

- Fewer false alarms appear,
- It can handle multiple types of anomalies, which significantly improves coverage,
- Risk-gate decisions are more stable because two independent perspectives reinforce each other.

The autoencoder in the KGK AI Transaction Blockchain is a neural network whose task is to restore the input to itself as best as possible.

- Input can be: X, which in this case is a transaction feature,
- Output: $\hat{X}$ is the input reconstructed by the network,
- The goal of the Autoencoder is to minimize the reconstruction error, $\hat{X} \approx X$.

This is an effective solution for anomaly detection because the neural network is mainly based on normal legitimate behavior, which is what it is trained to do.

This allows the neural network to learn to reconstruct normal patterns, but it reconstructs less well what is unusual for it. As a result, the reconstruction error increases, based on which the AI is able to detect the anomaly in the transaction. The use of this method is advantageous in the KGK AI Transaction Blockchain because an attacking transaction may be cryptographically valid (since the ECDSA signature is valid) but exhibit a behavioral pattern that deviates from normal in terms of amount, velocity, and partner. The structure of the Autoencoder consists of 3 elements, which are:

- Encoder. The encoder compresses the input into a lower-dimensional space, where X (dimension d) and Z (dimension k, $k < d$). This so-called compression process forces the neural network to learn the essential patterns of normal operation, not just copy them.
- Bottleneck, in other words, is the latent space, which is the hidden representation (Z). This is where what the neural network found to be relevant from the normal pattern is stored. If the input is unusual, the construction has a harder time fitting into Z.
- The decoder attempts to reconstruct the input. Z - $\hat{X}$

While learning, the KGK AI Transaction Blockchain uses the following objective function:

MSE (Mean Squared Error) (3)

$$loss = mean((X - \hat{X})^2) \quad (4)$$

The above formula can be used to accurately calculate the AnomalyScore when a new transaction is created where:

- FeatureVector: X,
- Autoencoder output: $\hat{X}$.

During the calculation, vectors of numerous normal transactions must be fed in, and the weights of the neural network must be adjusted so that typical patterns are well reconstructed during the process.

If the deviation were simply examined with an err solution, which indicates an error, it would not be a complete examination, since the scale of err depends on the:

- From feature normalization,
- From the model state and the ,
- Also, from data statistics.

The autoencoder parameters of the KGK AI Transaction Blockchain are as follows:

- Input dimension and feature set, where the input_dim parameter determines which features are fed into the neural network. If too few features are provided, the AI does not see enough patterns, resulting in more false negatives in the blockchain. On the other hand, if too many features are integrated, the training is more unstable, with more false positives and drift sensitivity.
- Architecture parameters, such as layers and neurons. The number of layers (num_layers) is very important in the encoder and decoder. In the case of a shallow layer, which includes only 1–2 layers, it is faster, more stable, and less overfit appears, which is an advantage, but the disadvantage is that the AI learns fewer nonlinear patterns. Deep layer means between 3 and 6 layers, which is stronger but more prone to overfitting and drift. In practice, 2–3 hidden layers are often enough for transactional anomalies.
- Width layer. Calculates how many neurons are needed per layer. (e.g. 32 - 16 - 8 - bottleneck). The effect of a large neural network is that it reconstructs everything more efficiently, even anomalies, thus reducing sensitivity. However, if the neural network is too small, it reconstructs incorrectly and therefore can give many false positives. The following sample is characteristic: Encoder: input_dim - 32 - 16 - latent_dim. Decoder: latent_dim - 16 - 32 - input_dim.
- Bottleneck size, or latent_dim. This is the most important parameter. The dimension of the middle bottleneck. Its adjustment ratio is latent_dim $\approx$ input_dim 3 or input_dim 4. This is the starting rule.

V. Experimental Evaluation

To validate the proposed KGK AI Transactional Blockchain, a proof-of-concept prototype and anomaly detection evaluation were implemented in a permissioned blockchain environment. The objective of the experiment was to determine whether AI-assisted anomaly detection can improve transaction security while maintaining acceptable blockchain performance.

The prototype used a Proof-of-Authority (PoA) blockchain with five validators. It integrated three anomaly detection mechanisms: Isolation Forest (iForest), One-Class SVM (OC-SVM), and an autoencoder neural network. The blockchain employed ECDSA P-256 signatures, AES-256-GCM encryption, and PBKDF2-based key derivation. The AI layer operated before transaction finalization and acted as a risk-gate capable of triggering quarantine, timelock, or manual approval procedures.

The experimental dataset contained approximately 128,000 synthetic enterprise blockchain transactions, including both legitimate and anomalous activities. The anomalous transactions simulated compromised keys, replay attempts, abnormal transaction velocity, unauthorized smart contract calls, and suspicious financial transfers. The following features were used during training and evaluation: transaction amount, sender velocity, recipient velocity, gas usage, contract method, transaction interval, and recipient frequency. All features were normalized before model training.

The autoencoder was trained exclusively on legitimate transactions using an unsupervised learning approach. The neural architecture contained an input layer of 32 neurons, two hidden layers (16–8–16), and a reconstruction output layer. The reconstruction loss was calculated using Mean Squared Error (MSE):

$$\text{Loss} = \frac{1}{n}\sum_{i=1}^{n}(X_i - \hat{X}_i)^2 \quad (5)$$

The Isolation Forest anomaly score was calculated using:

$$s(x) = 2^{-\frac{E[h(x)]}{c(n)}} \quad (6)$$

The OC-SVM decision function was defined as:

$$f(x) = \text{sign}\,(g(x)) \quad (7)$$

Transactions were classified as anomalous when the reconstruction error exceeded the threshold $\mu + 3\sigma$ derived from the normal transaction distribution. The final blockchain risk score combined the outputs of the three AI detectors using weighted averaging.

Table I summarizes the anomaly detection performance of the evaluated models.

TABLE I. Anomaly Detection Result

Model	F1-score	Recall	Precision	ROC-AUC
Isolation Forest	0.87	0.85	0.89	0.91
One-Class SVM	0.82	0.81	0.84	0.87
Autoencoder	0.92	0.91	0.93	0.96
Combined AI Layer	0.94	0.93	0.95	0.98

The combined AI layer achieved the best overall performance due to the complementary operation of the anomaly detection mechanisms. The autoencoder demonstrated the strongest standalone capability because it successfully learned nonlinear behavioral transaction patterns. The reconstruction error of anomalous transactions was significantly higher than that of legitimate operations, confirming the effectiveness of the neural anomaly detection layer.Blockchain performance overhead was also evaluated. The baseline PoA blockchain achieved 148 TPS with an average latency of 112 ms, while the KGK AI Blockchain achieved 131 TPS with 164 ms average latency. Although the AI layer introduced moderate computational overhead, the system significantly improved the detection of cryptographically valid but behaviorally suspicious transactions.

The experimental results demonstrate that integrating AI-based anomaly detection into a permissioned blockchain environment can substantially improve transaction security and trusted business communication while maintaining acceptable operational performance.

VI. Transactional Blockchain Trusted Business Communication in the KGK AI Transaction Blockchain

Trusted business communication in the KGK AI blockchain means that communication between business actors, such as orders, confirmations of performance, invoices, approvals,

contract amendments, and messages, is provably authentic, non-repudiable, cannot be modified unnoticed, and can be audited, while confidential content is selectively accessible.

In smart contract calls, communication is not just a message, but an operational intent. A contract call is represented as communication as follows:

- The participant requests the execution of payVendor(invoiceId) in the KGK AI Transaction Blockchain.
- This is followed by the signature,
- The AI neural network assesses the risk of the transaction,
- You can complete the transaction after the necessary checks.

During communication, the built-in AI Security Sentinel can indicate if a sudden burst of messages starts from a financial key, or if a large approval series is received towards a new recipient, or if an unusual time or method is used.

Security controls play an important role in the communication channel. Communication channel protection is necessary because a significant portion of real attacks occur at the ingress and network layers. These can include eavesdropping, man-in-the-middle, unauthorized submissions, bots, spam, DoS, Malicious payload, injection, message too large, Replays, duplications, Key and ID compromise and use of an old key. The gateway is the entry point of the KGK AI Transaction Blockchain. From this point:

- Receive transactions and business messages,
- Authenticates the sender,
- It checks policy, rate limit, and input validation.

It then passes the transaction on to the mempool and validators. The gateway, which is connected to the KGK AI Transaction Blockchain, only accepts certificates that:

- It comes from a pre-configured enterprise CA (Certificate Authority) chain,
- The chain is cryptographically valid (signatures are in order),
- The certificate is rooted in the appropriate trust store.

If this solution were not part of the KGK blockchain, then if any CA were accepted, anyone could generate any certificate for themselves, which would most likely violate the neural network's permissioned nature.

CONCLUSION

Due to their decentralization, blockchains are considered a more secure data storage solution than centralized solutions. However, this does not mean that the blockchain cannot be compromised, and suspicious financial transactions cannot be carried out within it. However, by using the built-in neural network and artificial intelligence together, transaction and party communication security can be significantly increased. To achieve this, it is necessary to use ECDSA (Elliptic Curve Digital Signature Algorithm) in the cryptographic layer of the blockchain, which creates a private key for the transaction, making it verifiable with the public key. The AES (Advanced Encryption Standard) 256-bit key in GCM mode enables the KGK AI blockchain to be encrypted and protected against manipulation, while PBKDF2 (Password-Based Key Derivation Function 2) repeats the hash-based operation with a very high number of iterations, which aims to make it difficult for an attacker to decipher the password. Connecting the KGK AI Transactional Blockchain with an AI-based smart chain is beneficial because it enables the smart contract to apply risk assessment solutions that can make transactions more secure.

In addition, the smart contract itself can detect suspicious transactions before they are permanently recorded in the blockchain. To achieve this, it is advisable to use artificial intelligence in both the blockchain and smart contracts. The purpose of applying AI is to filter out suspicious behaviors, thereby saving data in the blockchain that can communicate trust to users.

The prototype evaluation demonstrated that the combined AI detection layer achieved an F1-score of 0.94 and ROC-AUC of 0.98 while maintaining acceptable blockchain throughput. Future work will focus on real enterprise datasets, concept drift handling, and distributed validator-side AI inference.

REFERENCES

[1] A. Tick & N.Szabó-Harka, “Analysis of Information Security in a Corporate Environment–a Human Perspective,” In 2024 IEEE 22nd World Symposium on Applied Machine Intelligence and Informatics (SAMI) (pp. 469-474, 2024.

[2] V. Charles, A. Emrouznejad & T. Gherman, “A critical analysis of the integration of blockchain and artificial intelligence for supply chain,” Annals of Operations Research, 327(1), pp.7-47, 2023.

[3] A. Alghuried, M. Alkinoon, M. Mohaisen, A. Wang, C. Zou, & D. Mohaisen, “Blockchain security and privacy: Threats, challenges, applications, and tools,” Distributed Ledger Technologies: Research and Practice, 5(1), pp. 1-61,

[4] H. Han, R. K. Shiwakoti, R. Jarvis, C. Mordi & D. Botchie, “Accounting and auditing with blockchain technology and artificial Intelligence,” A literature review. International journal of accounting information systems, Elsevier, Volume 48, 100598, 2023.

[5] D. Liu, J. Zhang, Y. Wang, H. Shen, Z. Zhang, & T. Ye. (2025). ”Blockchain smart contract security: Threats and mitigation strategies in a lifecycle perspective,” ACM Computing Surveys, 58(4), pp. 1-34, 2025.

[6] G. Iuliano, & D. Di Nucci, “Smart contract vulnerabilities, tools, and benchmarks: An updated systematic literature review,” Journal of Systems and Software, 112788, 2026.

[7] T. Ashfaq, R. Khalid, A. S. Yahaya, S. Aslam, A. T. Azar, Alsafari, S., & I. A. Hameed, “A machine learning and blockchain-based efficient fraud detection mechanism,” Sensors, 22(19), 7162. 2022.

[8] T. Sowmya & E. M. Anita, “A comprehensive review of AI-based intrusion detection systems,” Measurement: Sensors, 28, 100827, 2023.

[9] T. Jiao, Z. Xu, M. Qi, S. Wen, Y. Xiang & G. Nan, “A survey of Ethereum smart contract security: Attacks and detection,” Distributed Ledger Technologies: Research and Practice, 3(3), pp. 1-28, 2024.

[10] N. Rane, S. Choudhary & J. Rane, “Blockchain and Artificial Intelligence (AI) integration for revolutionizing security and transparency in finance,” Available at SSRN 4644253, 2023.