DOI: https://doi.org/10.24297/ijct.v26i.9836

Main Title: Adaptive Cognitive Q-Learning–Based Security Model for Blockchain Protocols: Simulation and Experimental Evaluation on a Private Ethereum Network.

Rakotonanahary Fenitra [1], Robinson Hobihery Matio [2]

[1] Doctoral Student, Doctoral School of Engineering and Innovation Sciences and Technologies (STII), Laboratory of Cognitive Science and Application, University of Antananarivo, Madagascar.

[2] Associate Professor (Maître de conférences), Doctoral School of Engineering and Innovation Sciences and Technologies (STII), University of Antananarivo, Madagascar.

fenitrar07@gmail.com, matio.robinson@univ-antananarivo.mg

Abstract

The rapid growth of blockchain technologies has enabled decentralized applications based on smart contracts and distributed consensus. However, the increasing number of attacks exploiting protocol logic and network dynamics highlights the limitations of traditional, static security mechanisms. This study proposes an adaptive cognitive security model based on a Q-learning agent to enhance the protection of blockchain protocols. The agent is designed to analyze transaction behavior, assess risk levels, and dynamically select appropriate countermeasures. The proposed approach is evaluated through a dual experimental framework combining large-scale simulation using SimPy and execution on a private blockchain environment implemented with Ganache. Experimental results show a detection rate of approximately 70%, no observed false positives, a response time close to one second, and a very low operational gas cost. These results demonstrate that reinforcement learning can effectively improve the adaptability and responsiveness of blockchain security mechanisms while preserving network performance and economic viability. The study confirms the potential of cognitive and adaptive approaches for building more resilient and autonomous blockchain security systems.

Keywords: blockchain security, Q-learning, cognitive agent, adaptive defense, anomaly detection.

Introduction

Blockchain technologies have significantly transformed distributed systems by enabling decentralized trust and transaction integrity without relying on a trusted third party. Their increasing adoption in critical domains, particularly decentralized finance (DeFi) applications, has nevertheless exposed new security vulnerabilities. Attacks such as reentrancy, oracle manipulation, and flash loans demonstrate that blockchain security does not rely solely on cryptographic mechanisms but also on dynamic behaviors that are difficult to anticipate.

Current security approaches largely depend on manual audits, static rules, or post-incident patches. While these methods remain necessary, they show clear limitations when confronted with evolving and previously unknown attacks whose patterns are not always documented in advance. This context highlights the need for defense mechanisms capable of learning, adapting, and proactively responding to emerging threats without compromising the fundamental properties of blockchain systems.

In this regard, artificial intelligence, and more specifically reinforcement learning, provides a promising framework for enhancing the cybersecurity of distributed systems. By continuously observing network activity, a cognitive agent can adapt its decision-making process based on accumulated experience. When integrated into a blockchain protocol, such an agent can operate upstream of the consensus mechanism to analyze transactions before irreversible validation.

This paper proposes an adaptive cognitive model based on Q-learning, positioned between the mempool and the consensus process. The proposed approach aims to detect and mitigate abnormal behaviors in near real time while preserving network performance and decentralization. The model is evaluated through a dual experimental framework combining large-scale simulation and execution on a private Ethereum blockchain. The results highlight the relevance of jointly integrating adaptive learning mechanisms and blockchain architectures to build more autonomous and resilient security systems.

Materials and Methods

This study adopts an experimental methodology combining model design, large-scale simulation, and validation on a private blockchain. The objective is to evaluate the ability of a Q-learning–based cognitive agent to detect and mitigate abnormal behaviors in a blockchain environment while assessing its performance, responsiveness, and operational cost.

1. System architecture

The proposed system integrates a cognitive module positioned between the mempool and the consensus mechanism. This module acts as an analysis and decision layer capable of observing pending transactions, assessing their risk level, and applying appropriate actions prior to irreversible validation.

The architecture consists of three main components:

(i) a transaction data collection and analysis layer,

(ii) a Q-learning–based cognitive module,

(iii) an integration layer applying the agent's decisions to the consensus process.

The architecture is designed to enhance the blockchain's decision-making capabilities by monitoring and analyzing transactions in real time, enabling the application of countermeasures before final validation. The following figure (Figure 1) provides a visual representation of the general architecture of the adaptive cognitive model, illustrating the key components involved in data collection, decision-making, and action processes.

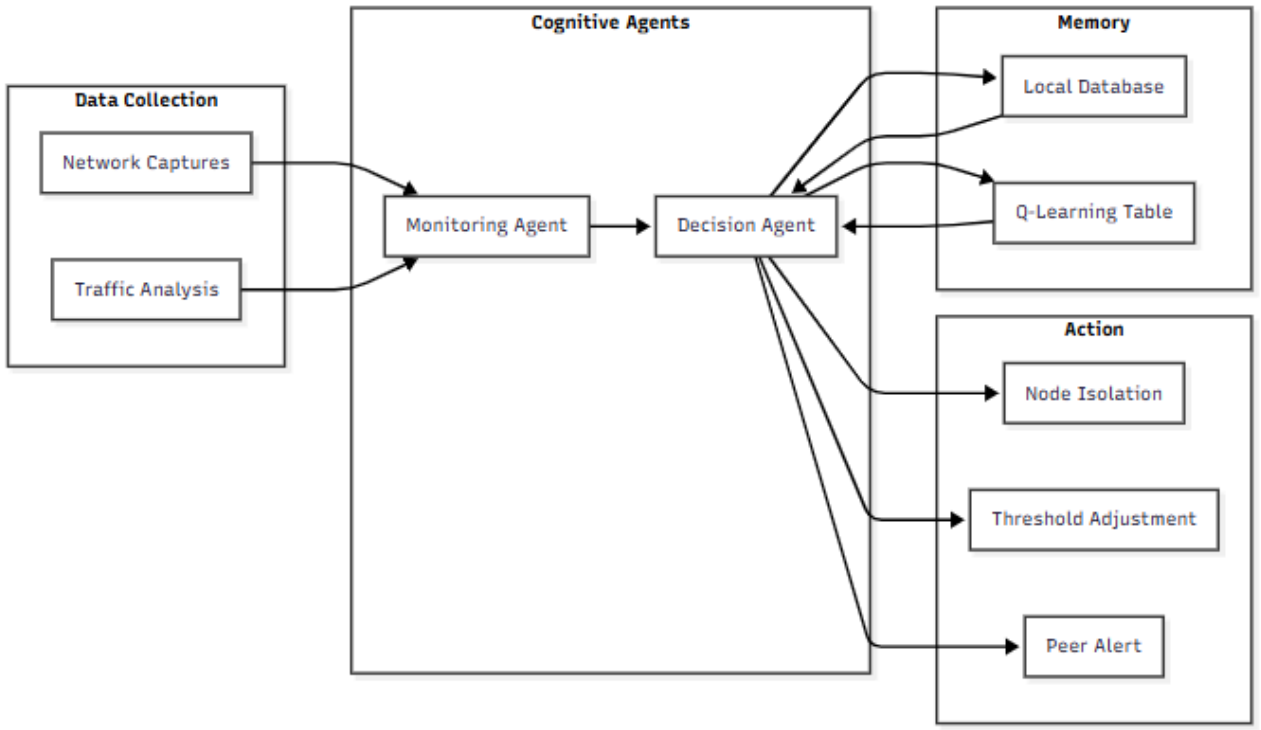

Figure 1: *General architecture of the adaptive cognitive model integrated between the mempool and the consensus.*

2. State and action representation

Each observed transaction is modeled as a state composed of several indicators, including contract call type, patterns associated with known attacks (reentrancy, oracle manipulation, flash loans), address behavior, and network context (gas price).

The cognitive agent can select among several actions: accepting the transaction, placing it under observation, isolating it for further analysis, or blocking it.

3. Reinforcement learning and Q-learning

Learning is based on the Q-learning algorithm, which enables the agent to estimate the value of actions associated with each observed state. The decision policy is progressively updated according to received rewards, which account for detection accuracy, false positives or negatives, response time, and gas cost.

The Q-learning algorithm updates the value of Q(s, a) as shown in the following central equation (Equation 1):

$$Q(s,a) \leftarrow Q(s,a) + \alpha\,[\,r + \gamma Q(s',a') - Q(s,a)\,] \quad (1)$$

Where:

- Q(s, a): estimated value of taking action a in state s
- α (alpha): learning rate
- r: reward associated with the action (good detection, false positive, cost, etc.)
- γ (gamma): discount factor
- max: maximum Q value for the next state-action pair.

4. Experimental environments

Two complementary environments are employed.

SimPy is used to rapidly simulate a large number of episodes and analyze the global behavior of the model at scale.
Ganache, a private Ethereum blockchain, is used to evaluate the system under near-real conditions, particularly in terms of latency and gas consumption.

```
(.venv) PS C:\xampp\htdocs\blockchain_ai_security\prototype> python main.py --no-ganache --duration 180
Mode d'exécution: Simulation
---- ÉCHANTILLON D'EXÉCUTION (5 dernières étapes) ----
{'temps': 175.0, 'action': 'isoler_adresse', 'recompense': 0.3, 'cout_eth': 0.05, 'note': 'Address isolated locally: None'}
{'temps': 176.0, 'action': 'ajuster_seuils', 'recompense': 0.0, 'cout_eth': 0.01, 'note': 'Threshold tightened -> 0.10'}
{'temps': 177.0, 'action': 'rien', 'recompense': 0.2, 'cout_eth': 0.0, 'note': 'Observation only.'}
{'temps': 178.0, 'action': 'isoler_adresse', 'recompense': 0.2, 'cout_eth': 0.05, 'note': 'Address isolated locally: None'}
{'temps': 179.0, 'action': 'rien', 'recompense': 0.2, 'cout_eth': 0.0, 'note': 'Observation only.'}

---- RÉSUMÉ ----
Mode: Simulation
Étapes: 179
Récompense moyenne: 0.227
Coût moyen: 0.018045 ETH
Coût total: 3.230000 ETH
Comptage des actions: {'rien': 89, 'alerte_pairs': 9, 'ajuster_seuils': 25, 'isoler_adresse': 56}
(.venv) PS C:\xampp\htdocs\blockchain_ai_security\prototype>
```

Figure 2: *Execution of the prototype in simulation mode (SimPy).* This figure provides a visual representation of the simulation environment used to evaluate the performance of the cognitive agent in detecting abnormal behaviors and assessing operational costs.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS                                   powershell - prototype

(.venv) PS C:\xampp\htdocs\blockchain_ai_security\prototype> python main.py --duration 180 --seed 42 --print-last 5 --export data/run_ganache_180s.csv
 Connexion à Ganache réussie
Mode d'exécution: Ganache
Solde initial: 99.8545 ETH
---- ÉCHANTILLON D'EXÉCUTION (5 dernières étapes) ----
{'temps': 175.0, 'action': 'isoler_adresse', 'recompense': 0.0, 'cout_eth': 0.00042, 'gas_used': 21000, 'note': 'Isolation simulée vers null; tx=0x93da16ea75948f9822bafef
1199ae09f9694f640733af093ade9d6653db39ce8; gas_used=21000; cost=0.00042000ETH'}
{'temps': 176.0, 'action': 'rien', 'recompense': 0.3, 'cout_eth': 0.0, 'gas_used': 0, 'note': 'Observation only'}
{'temps': 177.0, 'action': 'isoler_adresse', 'recompense': 0.0, 'cout_eth': 0.00042, 'gas_used': 21000, 'note': 'Isolation simulée vers null; tx=0x5cdda33ab912c203749a7e5
f64f8d4a897e75ad84c816b41d468d15244e0d497; gas_used=21000; cost=0.00042000ETH'}
{'temps': 178.0, 'action': 'isoler_adresse', 'recompense': 0.2, 'cout_eth': 0.00042, 'gas_used': 21000, 'note': 'Isolation simulée vers null; tx=0x42594b1e9309d35ca879a93
82e58782c9dff0ac6d7458f46bb0093d88043e123; gas_used=21000; cost=0.00042000ETH'}
{'temps': 179.0, 'action': 'rien', 'recompense': 0.2, 'cout_eth': 0.0, 'gas_used': 0, 'note': 'Observation only'}

Solde final: 99.8196 ETH
Coût total: 0.034860 ETH

---- ANALYSE DES COÛTS PAR ACTION ----
rien: 89 fois, coût total: 0.00000000 ETH, coût moyen: 0.00000000 ETH, gas total: 0, gas moyen: 0
isoler_adresse: 72 fois, coût total: 0.03024000 ETH, coût moyen: 0.00042000 ETH, gas total: 1512000, gas moyen: 21000
ajuster_seuils: 7 fois, coût total: 0.00000000 ETH, coût moyen: 0.00000000 ETH, gas total: 0, gas moyen: 0
alerte_pairs: 11 fois, coût total: 0.00462000 ETH, coût moyen: 0.00042000 ETH, gas total: 231000, gas moyen: 21000

---- RÉSUMÉ ----
Mode: Ganache
Étapes: 179
Récompense moyenne: 0.175
Coût moyen: 0.00019475 ETH
Coût total: 0.03486000 ETH
Gas total utilisé: 1743000
Gas moyen par transaction: 9737
Comptage des actions: {'rien': 89, 'isoler_adresse': 72, 'ajuster_seuils': 7, 'alerte_pairs': 11}
Blocs utilisés: 84
CSV exporté vers : C:\xampp\htdocs\blockchain_ai_security\prototype\data\run_ganache_180s.csv
```

Figure 3: *Execution of the prototype in real blockchain mode (Ganache).* This figure illustrates the real blockchain environment setup using Ganache, highlighting the integration of the cognitive agent for real-time decision-making and transaction validation.

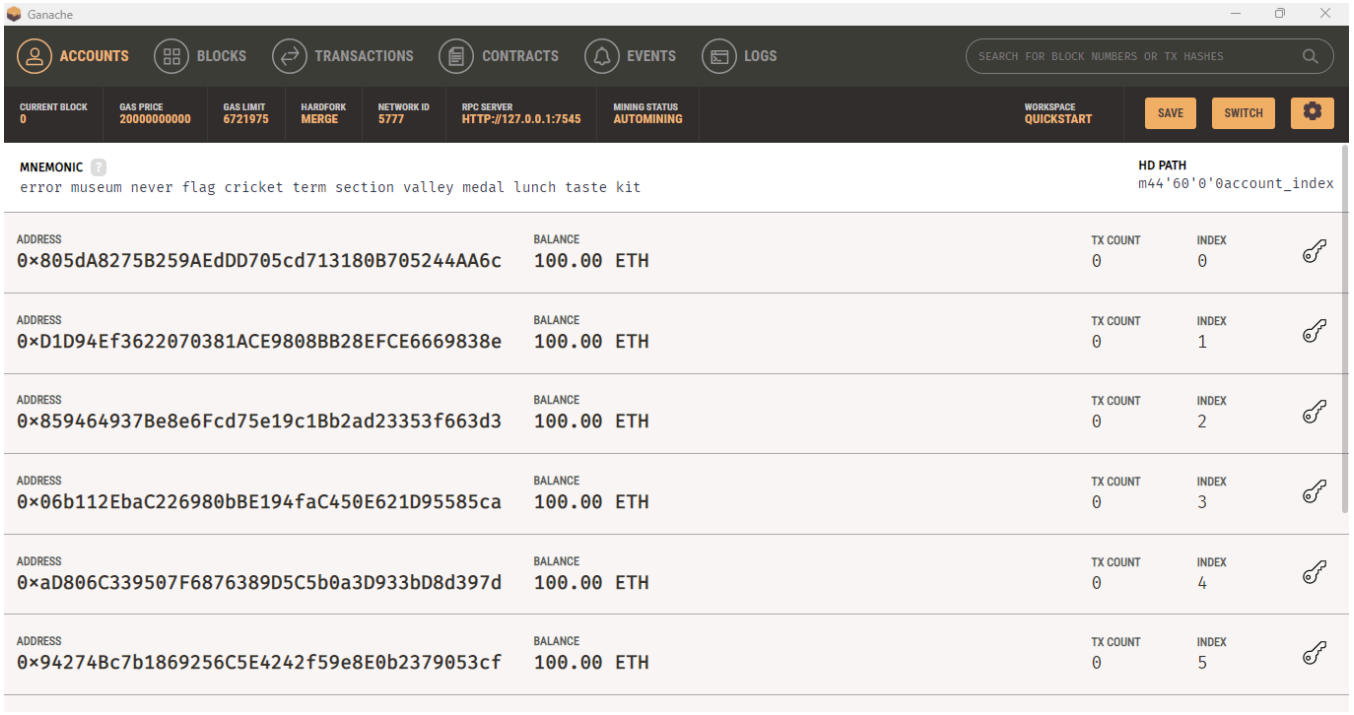

Figure 4: *Ganache environment setup for experimental evaluations.* This figure shows the setup of the Ganache environment, including the connection of accounts and the simulation of transactions on the private Ethereum blockchain for real-world testing.

5. Data collection and evaluation metrics

All agent decisions are logged in CSV format. The evaluated metrics include detection rate, false positives and false negatives, average response time, Q-table convergence, and operational gas cost. All scripts and generated data enable full reproducibility of the experiments.

Results and Discussion

This section presents the experimental results obtained from simulation and private blockchain execution, followed by a discussion analyzing the performance, robustness, and limitations of the proposed cognitive model.

1. Overall model performance

The model was evaluated in two distinct environments: large-scale simulation with SimPy and execution on a private Ethereum blockchain using Ganache.

Table 1 provides a comparison of the performance metrics, including detection rates, false positives, false negatives, total cost, average cost, and average time, across the two execution modes. The table highlights the differences in performance when the conditions are closer to a real-world blockchain environment.

Table 1: Comparison of performance metrics: SimPy vs Ganache

Mode	Detection Rate (%)	False Positives (%)	False Negatives (%)	Total Cost (ETH)	Average Cost (ETH)	Average Time (s)
SimPy	65.69	0.0	33.89	5.69	0.0238	1.0
Ganache	70.39	0.0	29.05	0.0349	0.000195	1.0

The results show a detection rate of **65.69%** in the simulation environment and **70.39%** in the Ganache environment. The improvement observed under near-real conditions suggests greater model stability when execution constraints are closer to those of an operational blockchain network.

No false positives were observed in either environment, which is a critical requirement for maintaining trust in a blockchain security system. However, false negatives remain relatively high (between **29% and 34%**), indicating that some attacks are still not automatically detected.

The average response time of the model is approximately **one second**, which is compatible with the time constraints of fast-paced attacks commonly observed in DeFi environments, such as flash loans.

Table 2 summarizes the robustness of the model's actions and results, comparing the performance of the cognitive agent in terms of isolated nodes, threshold adjustments, and the number of anomalies stopped and escaped in both the SimPy and Ganache environments. This comparison highlights the trade-off between cost and performance efficiency.

Table 2: Synthesis of robustness (actions and results).

Environment	No. of Isolations	No. of Adjustments	Total Cost (ETH)	Anomalies Stopped	Anomalies Escaped
Ganache	72	7	0.03024	40	39
SimPy	107	8	5.43	50	65

(i) Cost-performance efficiency: Ganache achieves a better cost-performance ratio, with only **0.03024 ETH** for 72 isolation actions compared to **SimPy**, which simulates a cost 180× higher. This demonstrates that **Ganache** is far more efficient in terms of cost while maintaining similar performance in terms of anomaly detection.

(ii) Anomalies stopped: SimPy stops more anomalies because it generates a larger number of transactions, which is expected in a simulation environment. However, Ganache shows a better cost-effectiveness ratio, as it requires fewer actions to stop similar anomalies, highlighting its operational efficiency.

(iii) Anomalies escaped: Both environments show numerous anomalies escaping, confirming the presence of false negatives, especially in complex scenarios where rapid adjustments are required.

2. Operational cost and economic feasibility

A key objective of this study was to assess the economic feasibility of embedding a cognitive agent within a blockchain protocol. The results reveal a significant contrast between simulated and real execution costs.

While SimPy produces high estimated costs due to its abstract nature, execution on Ganache demonstrates an extremely low real cost, with a total gas consumption of approximately **0.0349 ETH** across all episodes. The average cost per isolation action is below **0.001 ETH**, confirming that the proposed approach can be deployed without compromising the economic viability of the blockchain network.

3. Agent robustness and behavior

Analysis of the agent's actions indicates a predominance of address isolation decisions, combined with occasional threshold adjustments. In the Ganache environment, the model exhibits a better cost-efficiency ratio than in simulation, stopping a significant number of anomalies at a minimal operational cost.

However, some episodes reveal ineffective decisions, particularly when threshold adjustments fail to prevent ongoing attacks. These behaviors partially explain the persistence of false negatives and highlight the need for refining reward functions and expanding the action space.

4. Discussion and limitations

The results confirm that integrating a Q-learning–based cognitive agent can enhance blockchain protocol security in a proactive manner while preserving performance and decentralization constraints. A detection rate close to 70%, zero false positives, and very low gas costs demonstrate the feasibility of the proposed approach.

Nevertheless, the remaining false negatives reveal limitations in the current model configuration. These limitations are mainly related to the simplicity of the reward function and the tabular nature of Q-learning. More advanced approaches, such as deep reinforcement learning, fuzzy logic, or federated learning, could further improve threat anticipation and reduce undetected attacks.

Conclusions

This study investigated the integration of an adaptive cognitive agent based on Q-learning to enhance the security of blockchain protocols. By positioning the proposed module between the mempool and the consensus mechanism, the system introduces a proactive defense layer capable of analyzing transactions before irreversible validation.

The experimental evaluation, conducted through large-scale simulation and execution on a private Ethereum blockchain, demonstrates that the proposed approach can effectively detect abnormal behaviors while preserving network performance. The absence of false positives, a response time of approximately one second, and a very low operational gas cost confirm the practical feasibility of embedding a learning-based security mechanism within a blockchain environment.

Beyond performance considerations, this work highlights the relevance of combining reinforcement learning with blockchain architectures to address the dynamic nature of emerging threats, particularly in decentralized finance applications. While the current model achieves promising detection capabilities, the presence of false negatives indicates that further refinements are necessary to improve threat anticipation and decision accuracy.

Future work will focus on enhancing the learning strategy through more advanced reinforcement learning techniques, refining reward functions, and extending the model toward distributed and multi-agent implementations. The proposed framework also opens perspectives for application in other critical distributed systems, such as Internet of Things networks and decentralized infrastructures, where adaptive and autonomous security mechanisms are increasingly required.

Data Availability (excluding Review articles)

The data supporting the findings of this study were generated through controlled experiments conducted using simulation and a private Ethereum blockchain environment. All experimental data, including execution logs, performance metrics, and learning outputs, were automatically recorded during the experiments and stored in CSV format.

The complete source code of the proposed cognitive security model, including the Q-learning agent, simulation scripts (SimPy), smart contracts deployed on the private blockchain (Ganache), data analysis scripts, and raw experimental datasets, is publicly available to ensure full reproducibility of the results.

The repository also provides documentation describing the experimental setup and instructions for reproducing the simulations and blockchain-based evaluations.

All materials are accessible at the following public repository:

https://github.com/Fenitra07/blockchain_ai_security

References

Alharby, M., & Moorsel, A. van. (2017). Blockchain-based Smart Contracts : A Systematic Mapping Study. *Computer Science & Information Technology (CS & IT)*, 125-140. https://doi.org/10.5121/csit.2017.71011

Bashir, I. (2020). Mastering blockchain (2nd ed.). Packt Publishing.

Chainalysis. (2025). Blockchain security: Preventing threats before they strike. https://www.chainalysis.com

Choo, K.-K. R. (avec Dehghantanha, A., & Parizi, R. M.). (2020). *Blockchain Cybersecurity, Trust and Privacy*. Springer International Publishing AG.

Elrom, E. (2019). *The Blockchain Developer : A Practical Guide for Designing, Implementing, Publishing, Testing, and Securing Distributed Blockchain-based Projects*. Apress. https://doi.org/10.1007/978-1-4842-4847-8

Guru, A., Mohanta, B. K., Mohapatra, H., Al-Turjman, F., Altrjman, C., & Yadav, A. (2023). A Survey on Consensus Protocols and Attacks on Blockchain Technology. *Applied Sciences*, *13*(4), 2604. https://doi.org/10.3390/app13042604

IBM. (n.d.). What is blockchain security? https://www.ibm.com/think/topics/blockchain-security

ISO/IEC. (2022). ISO/IEC 27001:2022 – Information security management systems – Requirements. International Organization for Standardization. https://www.exactls.com/wp-content/uploads/2025/02/ISO_IEC-270012022-ed.3.pdf

Joint Task Force Interagency Working Group. (2020). *Security and Privacy Controls for Information Systems and Organizations* (Revision 5). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-53r5

Kamal, R. (Ed.). (2020). *Handbook of research on blockchain technology*. Academic Press.

Kamhoua, C. A., Njilla, L. L., & Shetty, S. (Éds.). (2019). *Blockchain for distributed systems security*. Wiley-IEEE. https://doi.org/10.1002/9781119519621

Kshetri, N., Pandey, P. S., & Ahmed, M. (2024). *Blockchain Technology for Cyber Defense, Cybersecurity, and Countermeasures : Techniques, Solutions, and Applications* (1re éd.). CRC Press. https://doi.org/10.1201/9781003449515

Maleh, Y., Baddi, Y., Alazab, M., Tawalbeh, L., & Romdhani, I. (Éds.). (2021). *Artificial intelligence and blockchain for future cybersecurity applications*. Springer International Publishing AG.

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. https://bitcoin.org/bitcoin.pdf

OWASP Foundation. (2025). OWASP smart contract top 10. https://owasp.org/www-project-smart-contract-top-10/

Zhang, R., Xue, R., & Liu, L. (2020). Security and Privacy on Blockchain. *ACM Computing Surveys*, *52*(3), 1-34. https://doi.org/10.1145/3316481

Supplementary Materials

Supplementary materials associated with this study include the source code of the proposed cognitive security model, simulation scripts, smart contracts deployed on a private Ethereum blockchain, and experimental datasets generated during the evaluation.

These materials provide additional technical details that support the reproducibility of the experiments and the validation of the reported results. The complete supplementary material is provided as a compressed archive and is publicly accessible through the project repository referenced in the Data Availability section.

Conflicts of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Funding Statement

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Acknowledgments

The authors would like to thank the Doctoral School of Engineering and Innovation Sciences and Technologies (STII) of the University of Antananarivo for providing the academic environment and support necessary for this research. The authors also acknowledge the Laboratory of Cognitive Science and Application for the technical and scientific framework that facilitated the development and evaluation of the proposed model.