

BC-XAIA: An Intelligent Framework for Securing Autonomous Software Using Blockchain Technology and Smart Contracts

Hebat Allah Adel^{1,*}, Sayed AbdelGaber¹ and Wessam H. El-Behaidy^{1,2}

¹ Department of Computer Science, Faculty of Computers and Artificial Intelligence, Capital University, Cairo, Egypt.

² Faculty of Informatics and Computer Science, British University in Egypt, El-Sherouk, Egypt.

* Corresponding author: Hebat Allah Adel (e-mail: Hebaadel57@gmail.com).

Contributing authors: Sayed AbdelGaber (sgaber14@gmail.com); Wessam H. El-Behaidy (W_behaidy@fci.capu.edu.eg, Wessam.elbehaidy@bue.edu.eg.).

Received: 17 June 2026 **Accepted:** 15 October 2026 **Published:** 18 October 2026

DOI: 10.15849/ijasca.181 **License:** Open access under CC BY 4.0. Authors retain copyright.

Publisher: Al-Zaytoonah University of Jordan (ZUJ), Amman, Jordan.

Cite this article: H. A. Adel, S. AbdelGaber, and W. H. El-Behaidy, "BC-XAIA: an intelligent framework for securing autonomous software using blockchain technology and smart contracts," *Int. J. Advance Soft Compu. Appl.*, vol. 18, no. 3, pp. 561-576, 2026, doi: <https://doi.org/10.15849/ijasca.181>.

ABSTRACT

Rapid deployment of autonomous software in enterprise environments has raised serious security concerns regarding data integrity and system transparency. While blockchain provides decentralized protection, combining it with explainable AI to secure autonomous decisions remains understudied. This paper introduces BC-XAIA, an intelligent framework that integrates Ethereum smart contracts, Explainable AI (XAI), and Agile practices. The framework creates immutable logs for system actions while using SHAP and LIME to generate clear, real-time explanations for automated outputs. BC-XAIA was evaluated through an HR management case study across 15 functional and non-functional test cases. Usability testing with the System Usability Scale (SUS) resulted in an average score of 86%, placing the system in the excellent category for user satisfaction and integration. Overall, BC-XAIA demonstrates a practical way to combine blockchain security with transparent AI decision-making in real-world software.

Keywords: Blockchain; Smart Contracts; Explainable AI; Agile; Autonomous Software.

MSC Codes: 68T01, 68M14, 90B50.

1 Introduction

Autonomous software systems are growing fast in industries. This growth brings critical security concerns. It is important to find ways to protect this technology. Blockchain technology has become a solution for security problems in various areas. Using it to protect autonomous software systems is still unexplored [1]. This study examines how autonomous software systems can be secured to address security problems and detect threats quickly. By answering this question, this study helps make autonomous software systems more secure and reliable. This can increase people's confidence in using these systems in life.

Blockchain technology has demonstrated potential in supporting automation, security, and transparency in digital systems. At the same time, Artificial Intelligence (AI) is being increasingly adopted to help systems make decisions based on data. However, many AI models are still considered black boxes because it is often difficult to understand how they reach their decisions. When models lack interpretability, ensuring fairness and accountability becomes difficult [2].

To address these challenges, this study proposes BC-XAIA, an intelligent framework that integrates blockchain technology, Explainable Artificial Intelligence (XAI), and agile methodologies. The framework uses Shapley Additive exPlanations (SHAP) and Local Interpretable Model-agnostic Explanations (LIME) to make AI-based decisions easier for stakeholders to understand. It also follows an agile mechanism that supports iterative updates based on user feedback and changes in real-time insights. The proposed system was validated through test cases conducted by an HR professional, covering both functional and non-functional requirements.

Research Objectives:

- (1) Developing a technique to verify the software's actions and identify any anomalies or suspicious behaviour.
- (2) Developing a framework or model that integrates blockchain technology into the data management of autonomous software systems to improve security.
- (3) Providing recommendations and guidelines for the practical implementation and deployment of the intelligent framework in real-world scenarios.
- (4) Identifying potential limitations and areas for improvement in an intelligent framework.

This study presents the design, implementation, and evaluation of BC-XAIA, demonstrating how emerging technologies can be effectively integrated to enhance and improve current practices. BC-XAIA differs from existing integrations, including our previous BC-HRM. In BC-HRM, the smart contract was relatively static. On the other hand, BC-XAIA adopts a dynamic proxy smart contract that supports controlling updates to the system rules. The framework also establishes a two-level explainability mechanism by combining the global explanations generated using SHAP and local explanations generated using LIME. Also, the smart contract participates in the verification process by checking the predefined conditions related to the XAI outputs before allowing subsequent system actions. Therefore, the framework extends the role of blockchain from data integrity and traceability toward supporting the verification and accountability of AI-driven decisions.

The rest of this paper is organized as follows: section 2 examines the use of smart contracts and their role in secure autonomous software systems; section 3 summarizes the relevant literature on blockchain and smart contracts for securing autonomous and intelligent software systems; section 4 presents the details of the analysis and design of BC-XAIA; section 5 reports the evaluation of the proposed BC-XAIA; and finally, section 6 discusses the conclusion and highlights potential directions for future work.

2 Smart Contracts

Smart contracts are self-executing digital agreements stored on a blockchain that are designed to automatically enforce the terms of an agreement once the predefined conditions are met. These cryptographic constructs replace traditional legal enforcement mechanisms with code-based logic, shifting trust from institutions to decentralized networks [3]. Platforms ensure immutable and predictable execution of tasks [4].

2.1 Ethereum and Its Role in Secure Autonomous Software Systems

Ethereum serves as a foundational ecosystem for autonomous software systems because of its robust decentralized framework [5]. Smart contracts allow autonomous software to perform tasks such as data validation, access control, and transactional processing without continuous human intervention. This can help reduce manual work and limit operational errors [6]. Its underlying consensus mechanism eliminates the need for trusted intermediaries, thereby strengthening the overall system resilience [7]. Once deployed, the smart contract code and transaction history remain immutable, establishing clear audit trails. Additionally, Ethereum wallets function as decentralized identities that streamline role-based access control. The platform's compatibility with token standards also enables the creation of incentive mechanisms and reputation systems that can autonomously reward or penalize participants based on predefined criteria [8]. When combined with blockchain oracles, smart contracts can receive information from external data sources. This allows real-world events to trigger actions on the blockchain, making software systems more autonomous and better able to respond to changes in their surrounding environment. Despite challenges related to scalability and smart contract vulnerabilities, Ethereum remains a foundational technology for designing secure, transparent, and autonomous systems in domains such as recruitment, supply chain, and finance [9].

3 Related Work: Blockchain and Smart Contracts for Securing Autonomous and Intelligent Software Systems

The integration of blockchain technology and smart contracts has emerged as a key approach to securing autonomous software systems. Their use can also support autonomous systems in domains where trust is important. To enhance data security in intelligent architectures, Li et al. [10] proposed a blockchain-based framework for 6G wireless networks. Although their study focused on indoor positioning and autonomous vehicle use cases, it successfully demonstrated the resistance of blockchain to data tampering. Highlighting a critical mechanism for securing the distributed data repositories required in modern automated systems.

Similarly, maintaining data integrity during real-time monitoring is vital to the autonomous framework. Krishna and Tyagi [11] suggested integrating Intrusion Detection Systems (IDS) with blockchain for Intelligent Transportation Systems (ITS). Their approach uses blockchain to protect stored data from unauthorized changes. This is useful for auditing automated decisions and protecting sensitive records.

In the context of operational accountability, smart contracts have been used to automate and secure the execution of independent workflows. Muram and Javed [12] introduced a blockchain-based risk management system for drone operations. The system uses smart contracts to improve security and supports the integration of machine learning for more adaptive and verifiable decisions.

Kamble et al. [13] examined Ethereum and Bitcoin platforms in the context of autonomous systems. Their study focused on improving transparency and decentralization. Jain et al. [14] examined different models using blockchain in autonomous systems. The findings highlighted its potential to improve user experience, operational trust, and scalability.

Gupta et al. [15] supported the critical need for robust cryptographic protocols to secure system communications. An extensive risk assessment of connected vehicles was conducted to propose targeted countermeasures. Dakić and Živković [16] also examined legal, ethical, and multidisciplinary issues in building autonomous software stacks. Highlighting the need to use transparent and accountable ledger technology to address stakeholders' concerns.

Furthermore, data privacy and resistance to cybersecurity threats remain foundational pillars of any autonomous framework. SECVT was developed by Tyagi et al. [17] using a decentralized architecture through cryptographic techniques. To reduce cybersecurity risks, Hataba et al. [18] studied various methods of layering security defences. Showing the use of how distributed technologies could effectively

prevent single-point-of-failure issues.

Security and audit logging are critical when automated systems interact with external assessment bodies. Parlak [19] developed a blockchain-based system for accident evidence management. It leveraged smart contracts and an immutable record to improve the transparency of multiparty processing. To achieve scalability and cost-effective ledger storage, Rana et al. [20] introduced a decentralized model for securing digital evidence. Smart contracts were used on a layer 2 blockchain, ensuring end-to-end data immutability and trackability.

Zubaydi et al. [21] provided a systematic review of blockchain applications in software-driven systems. Highlighting its potential in strengthen trust and transparency. Particularly in environments where IOT and autonomous systems share similar architectural characteristics. Ahmad et al. [22] investigated the use of machine learning with blockchain to improve system resilience. The findings showed that secure validation of data can contribute to more effective anomaly detection. However, existing decentralized architectures are vulnerable. Guo and Yu [23] reviewed the security foundations and limitations of current consensus mechanisms. New directions were indicated for refining the cryptographic frameworks. Kharche et al. [24] discussed the implementation of blockchain in IoT-driven environments, emphasizing that scalability and cost efficiency remain open challenges that must be addressed when building intelligent software infrastructures.

Unlike previous studies and implementations such as BC-HRM, where blockchain mainly serves as a record of transactions. BC-XAIA brings these elements together within a single architecture. The framework combines dynamic proxy smart contracts with SHAP and LIME explanations to allow the decision-making process to be updated when needed, while keeping the related information traceable and verifiable throughout the process.

Despite the growing use of autonomous software systems, several research gaps remain. Particularly in addressing security vulnerabilities and supporting timely detection and response to potential threats. Blockchain has shown potential for improving data security and trust. However, its combined use with smart contracts and artificial intelligence for addressing security issues during system operation remains relatively underexplored. This highlights the need for integrated security approaches. Such approaches should combine secure data management, automated execution, and intelligent security mechanisms.

4 BC-XAIA Proposed Framework

The proposed BC-XAIA framework integrates three core pillars: blockchain technology, Explainable Artificial Intelligence (XAI), and the agile methodology, as shown in Fig. 1. Combining these components provides a secure and transparent architecture that can be adjusted to meet the needs of autonomous software systems. XAI provides model explanations that allow stakeholders to verify predictions. Finally, the agile methodology introduces iterative development and continuous feedback loops to maintain architectural adaptability.

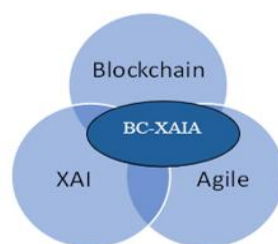


Figure 1. Proposed framework for discipline.

4.1 BC-XAIA Components

As shown in Fig. 2, the framework comprises seven primary components: user, smart interface, blockchain, smart contracts, database, XAI, and agile. The smart interface serves as the primary interaction point between users and the autonomous application.

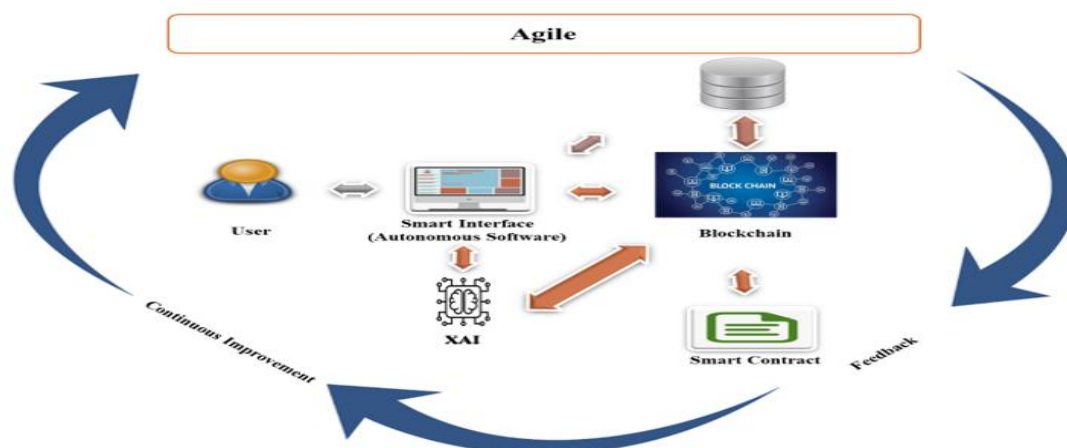


Figure 2. BC-XAIA System Components.

The interface also uses Explainable AI (XAI) techniques. These techniques make system decisions easier to understand and provide clearer explanations for the users involved. The smart interface is

connected to the blockchain network. This provides a secure and decentralized environment for all the important processes performed by the system. Smart contracts are used in blockchain to automatically execute workflows depending on certain conditions. It also minimizes human intervention. The entire framework was integrated into an agile development cycle to provide flexibility and constant improvement.

4.2 BC-XAIA Workflow and Processes

Fig. 3 emphasizes how different technologies, such as blockchain, XAI, smart contracts, and agile, collaborate. And how these technologies provide transparency to users. It also allows development in an iterative process. Fig. 4 illustrates the mechanisms used for collecting, validating, and storing the data. Collectively, these elements represent a complete framework for creating an intelligent and autonomous software system.

As illustrated in Fig. 3, the operation of the workflow process of the BC-XAIA shows the relationships between the key components involved in securing an autonomous software system. The workflow process starts when a user interacts with the system using the smart interface (Autonomous Software). Through the smart interface, predefined workflows governed by smart contracts are triggered to ensure that business rules and logic are securely enforced on the blockchain. On the other hand, AI decision-making occurs, and its results are processed through XAI elements.

The database module ensures that the data are stored in a secure and structured manner and are synchronized with the blockchain, to ensure secure and reliable storage of the data. All important transactions and interactions are stored on the blockchain, ensuring that the system's interactions are tamper-proof and verifiable. Adopting agile principles served as the core foundation for constructing the entire framework. Through an iterative loop connecting direct user feedback with continuous refinement. The architecture can adapt to changing operational needs. It also provides a secure and explainable environment for autonomous software systems.

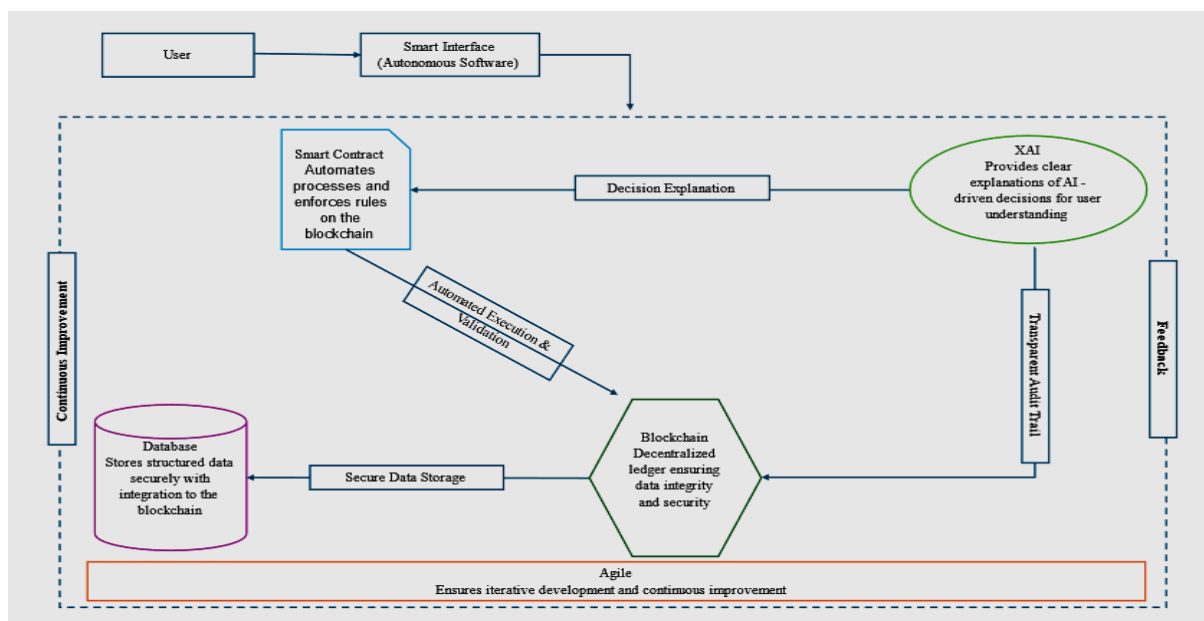


Figure 3. Operational workflow of BC-XAIA.

Figure 4 shows the data lifecycle and security mechanisms of the BC-XAIA framework. The process starts with iteration planning and data collection by the autonomous software. The collected data then passes through several processing stages. Transactions are grouped into blocks, and each block is given a unique cryptographic signature. These blocks are linked together to form a chain, making unauthorized changes difficult. Distributed network nodes then verify the blocks using a consensus mechanism before adding them to the blockchain. This ensures that only valid transactions are recorded.

The smart contract executes the required business processes and applies access restrictions. It also interacts with the XAI modules when needed. Access controls and system logs help protect and track critical operations. The XAI component provides understandable explanations for the decisions made by the system. These explanations help end users and external auditors monitor its behavior. In addition, the database runs in parallel with the blockchain to store structured data. The explanations produced by the XAI module also help users understand how the system behaves and provide better traceability of its actions. These are two essential aspects required for the implementation of an autonomous system in different sectors such as recruitment and finance.

The feedback loop ensures ongoing improvement and enhancement of the system using data.

To handle transactions efficiently, the system uses a Proof-of-Stake (PoS) consensus mechanism instead of traditional Proof-of-Work. In our setup, validator nodes confirm blocks based on staked assets. minimizes energy use, speeds up execution, and cuts transaction fees. When tested on our local Ethereum network, this mechanism allowed smart contracts and XAI metadata to record updates instantly. Making it ideal for fast, real-time enterprise systems.

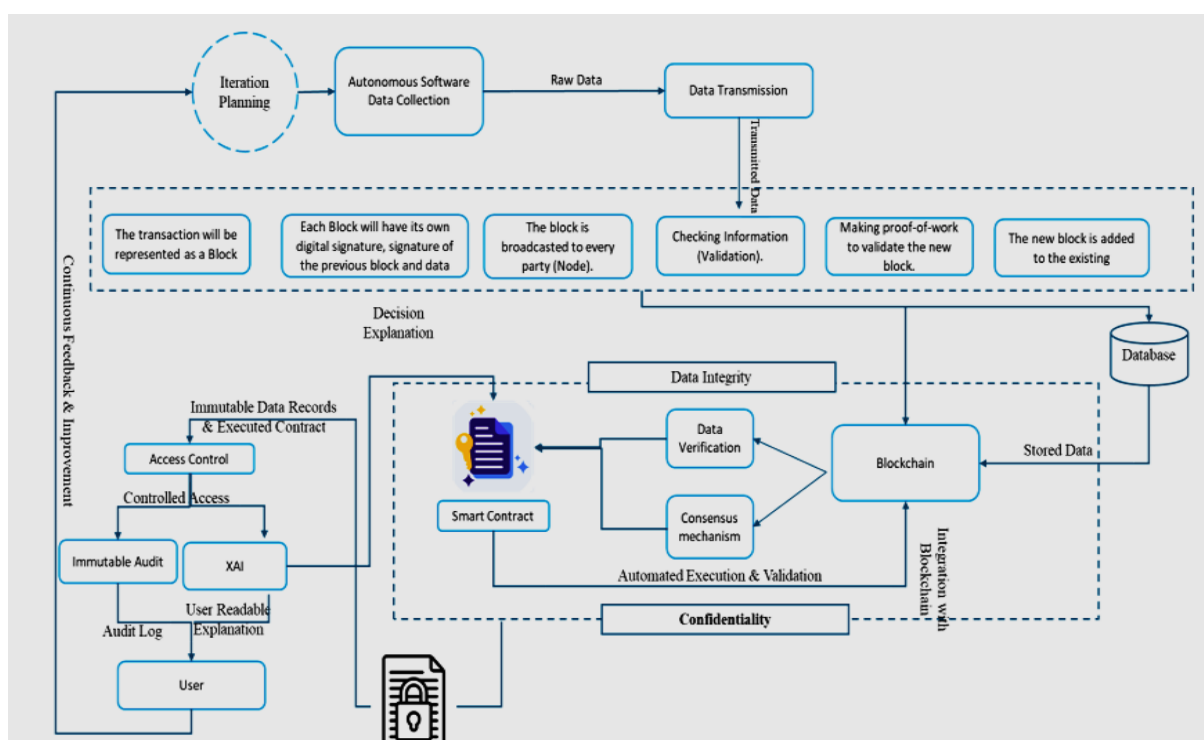


Figure 4. Processes of the proposed BC-XAIA model.

5 BC-XAIA Implementation

To implement the proposed BC-XAIA framework, technical deployment, architectural tools, and components were realized for validation.

5.1 Smart Contract Execution

Smart contracts deployed on the Ethereum blockchain manage the decentralized logic of the BC-XAIA framework. They were written in Solidity. These contracts automate workflow execution and remove the reliance on centralized intermediaries. Network nodes execute contracts peer-to-peer to maintain state consensus and provide immutable logs [25].

Ethereum was selected for prototyping, because of its mature Virtual Machine (EVM) and broad developer support. Among the tested IDEs (Ganache, Truffle, and Remix). Remix was chosen for contract creation and compilation because of its debugging capabilities and environment integration.

The primary functional modules in the Solidity architecture include the following operations.

- (1) Fallback Function: Implemented as a security countermeasure to protect the smart contract from attacks.
- (2) Set Contract: Executes cryptographic signature verification. It authenticates user addresses, organization identifiers, and contract IDs to prevent forgery and maintain auditable access control.
- (3) Set User: Performs activation for the user to be added to the blockchain.
- (4) Count Users / Count Users C: Administrative functions designed to compute active system entities and track the quantitative boundaries of active organizational agreements.
- (5) Filter: Provides algorithmic categorization of users based on multidimensional performance metrics and historical evaluation attributes.
- (6) Get Users: Retrieves and orchestrates participant addresses to govern multiparty transactions under defined policy constraints.
- (7) Login: Secure verification of access credentials is conducted to protect confidential data and mitigate unauthorized infiltration.
- (8) Track User/Track User contract: Monitors users and contract state changes.

5.2 BC-XAIA Smart Interface

BC-XAIA was designed using the Microsoft Visual Studio development suite. The integration between the smart interface and the Ethereum blockchain was facilitated using the Azure Blockchain Development extension in Visual Studio Code. It handles the compilation, building, and deployment pipelines for Solidity smart contracts.

The front-end framework controls access based on user roles. It provides different interfaces according to each stakeholder's permissions. In the Human Resource Management (HRM) case study, the system includes several modular interfaces. It includes localized subsystems for authentication, employee data management, vacancies, and algorithmic criteria configuration. It also includes explanation and stakeholder feedback collection.

5.3 BC-XAIA Database

To validate the scalability of the framework, a hybrid storage architecture was implemented. The relational operational data were managed off-chain. The critical transactional hashes were anchored to the blockchain. The off-chain data repository was designed using Microsoft SQL Server, structured across 15 interconnected relational tables, as illustrated in the Entity Relationship Diagram (ERD) in Fig. 5.

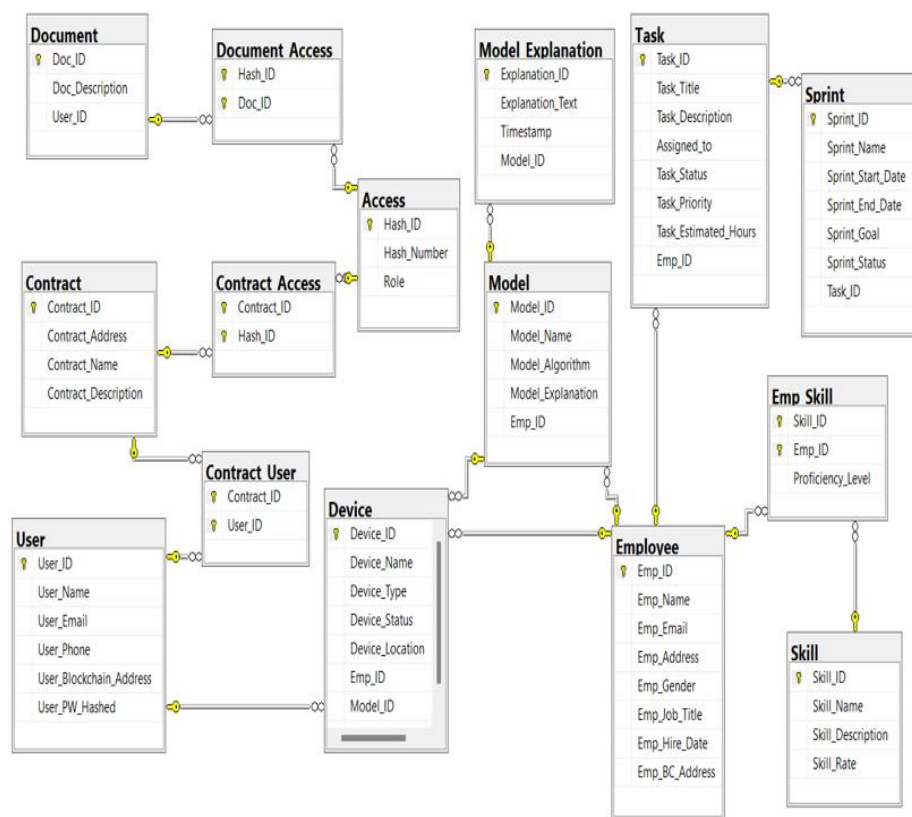


Figure 5. Entity relationship diagram for the BC-XAIA database.

The architecture of the database uses Role-Based Access Control (RBAC) for data governance. Common employees have access only to privileged Data Manipulation Language (DML) commands that pertain to them. HR members have privileges that allow them to perform administrative functions such as insertions, modifications, and deletions within the context of department, candidate, and employment datasets.

5.4 BC-XAIA Explainable Artificial Intelligence

BC-XAIA reduces on-chain storage and computational costs. by running machine-learning pipelines off-chain while recording decision proofs on-chain. The models were trained locally using candidate datasets. SHapley Additive exPlanations (SHAP) provide global feature importance across the dataset, whereas Local Interpretable Model-agnostic Explanations (LIME) provide local explanations for individual predictions.

To evaluate the framework in a recruitment case study, a Random Forest classifier was trained on a dataset comprising 1,500 records (1,200 for training and 300 for testing). The Random Forest model achieved 93% predictive accuracy, 0.93 precision, 0.83 recall, an F1-score of 0.88, and an AUC-ROC of 0.94 [26]. SHAP was used to calculate global feature importance across the dataset to provide insights to key stakeholders regarding top contributing features such as Education Level, Interview Score, Personality Score, and Skill Score. In contrast, LIME was employed for local instance-level interpretability. Detailed candidate-level XAI visualizations and feature weight metrics are fully documented in our prior work [26].

To bridge the gap between off-chain AI predictions and decentralized security, a Prediction Contract was developed using Solidity. The smart contract records the cryptographic proof of each decision on the blockchain. This allows automated recruitment decisions to be verified and traced without changing the recorded data or exposing sensitive personal information off-chain.

The BC-XAIA framework uses an Agile approach with repeated development cycles. This allows the system and its models to be updated as requirements change. Security and system performance are also improved throughout the sprint cycles, as shown in Fig. 6.

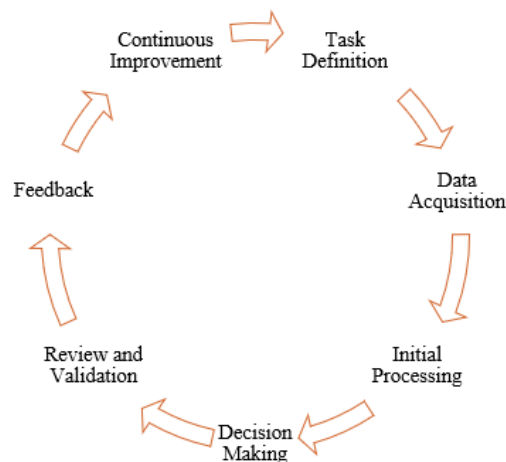


Figure 6. BC-XAIA's Agile Approach for Enhancing Efficiency and Continuous Improvement.

The lifecycle is initiated with iterative task definition and collaborative sprint planning, establishing the functional scope and data parameters for the autonomous system. During the data acquisition and processing phases, the system collects and filters data from user interactions and historical ledgers. The machine learning model undergoes continuous retraining within these iterative loops, allowing the system to collect feedback and adapt to new operational conditions.

The resulting dynamic outputs and explanation metrics were continuously delivered to HR stakeholders. This continuous feedback loop is fed back into system retrospectives, triggering automatic smart contract adjustments or model optimizations. Consequently, the integration of agile principles ensures that the BC-XAIA maintains performance, transparency, and contextual responsiveness in dynamic enterprise environments.

6 BC-XAIA Evaluation and Results

A structured empirical evaluation for BC-XAIA was executed. To rigorously evaluate the functional reliability, operational security, and user experience. The assessment methodology follows systematic phases, starting with setting the test plan and ending with standardized usability auditing.

6.1 Experimental Setup and Test Execution

The evaluation started by defining the testing scope and the criteria for starting and completing the tests. 15 functional and non-functional test cases were then prepared. These tests covered key system operations, including cryptographic user authentication, smart contract state changes, database synchronization, and the real-time display of SHAP and LIME explanations. The test cases were executed within a controlled enterprise deployment environment. The functional tests included boundary inputs and checks for deterministic execution. These tests verified that the Solidity fallback functions and signature verification triggers worked as expected without causing state failures or security issues.

Implementation Setup and Experimental Environment

Smart contract orchestration was compiled and executed within the Ethereum Remix IDE environment. Machine learning workflows encompassing model training, cross-validation, and explainable AI (XAI) feature extraction were hosted on an Intel Core i7 architecture equipped with 16 GB RAM. Running Python 3.10 and Scikit-Learn. Empirical validation utilized a 1,500-instance recruitment dataset

evaluated via a 5-fold cross-validation scheme. Table 1 details the 15 executed test scenarios alongside their respective inputs, expected performance metrics, and empirical outcomes.

Table 1. Functional and Non-Functional Test Case Matrix and Operational Reliability Assessment.

Test ID	Scenario / Operational Pathway	Input Data	Expected Result	Actual Result	Status
TC-01	Cryptographic Authentication	Recruiter login credentials	Valid credentials & session initialized	Session successfully initialized	Pass
TC-02	Candidate Profile Submission	Candidate attribute (ID, Address, Rate, Skills, performance)	Profile validated & stored on-chain via smart contract	Profile successfully validated and committed to blockchain	Pass
TC-03	Smart Contract Status Binding	Organization ID, Contract ID, Signature Status	Contract status recorded & saved on-chain	Status successfully recorded via smart contract	Pass
TC-04	On-Chain Candidate Query	Authorized User Request	Employee/Candidate address retrieved from smart contract	Address successfully queried and displayed	Pass
TC-05	Candidate Attribute Filtering	Candidate Address & Criteria Threshold (greater than or equal 5)	Filtered profile details returned via smart contract	Candidate metrics accurately retrieved and filtered	Pass
TC-06	On-Chain Vacancy Query	Job Seeker Request	Available vacancies queried & retrieved from smart contract	All published vacancies correctly displayed	Pass
TC-07	Smart Contract Binding & Status Validation	Organization ID, Contract ID, Signature Status	Contract metadata & signature status committed on-chain	Contract status successfully saved via smart contract	Pass
TC-08	User Profile Data Store	Candidate profile metrics (ID, Rating, Skills)	Employee record updated on blockchain	Record successfully created via smart contract	Pass
TC-09	Active Vacancies Query	HR/Applicant Query Request	Existing vacancies retrieved via smart contract	All active job postings retrieved and rendered	Pass
TC-10	On-Chain Vacancy Creation	New vacancy data payload	Vacancy created and saved on-chain	New vacancy successfully committed via smart contract	Pass
TC-11	Employee Contract Aggregation	Organization ID / Query Request	Total count and range of assigned contracts retrieved	Contract metrics successfully aggregated and displayed	Pass
TC-12	Total Candidate Count Retrieval	System Aggregation Query	Total registered employee count returned	Employee tally accurately queried and displayed	Pass
TC-13	Skills Repository Query	Skill Set Read Request	Available skills retrieved via smart contract	Skill matrix successfully rendered	Pass
TC-14	Skill Definition & Store	Skill metadata (Name, Description, Rating)	Skill added and committed on-chain	New skill successfully registered via smart contract	Pass

TC-15	On-Chain Employee Onboarding	Employee metadata & address	New employee record committed on-chain	Profile successfully created via smart contract	Pass
-------	------------------------------	-----------------------------	--	---	------

6.2 Security, Integrity, and Threat Evaluation

To validate the security of the BC-XAIA framework, a threat analysis was conducted to examine how the system responds to common vulnerabilities in decentralized environments. The analysis focused on the system's resilience under simulated operational conditions. The framework uses cryptographic hashing, Ethereum-based role access control, and smart contract verification to protect data integrity. Table 2 presents the identified threat scenarios, the corresponding mitigation mechanisms, and the results of the conducted tests.

Table 2. Security Threat Analysis and Attack Mitigation Matrix.

Threat Vector	Attack Mechanism	BC-XAIA Mitigation Strategy	Experimental Result	Status
Data Tampering and Forgery	Unauthorized alteration of evaluation records and user profiles.	Cryptographic signature verification via Set Contract function combined with immutable log anchoring in Set user function.	Hashing mismatches detected instantly; unauthorized state changes rejected.	Pass
Reentrancy and Fallback Exploits	Malicious contracts invoking fallback execution to drain/disrupt state.	Explicit implementation of Fallback Function protection mechanisms and strict checks effects interactions.	Simulated recursive contract calls rejected without system state compromise.	Pass
Unauthorized Access	Identity spoofing to access confidential evaluation features.	Multi-tier credential verification in Login function combined with get Users function policy constraints.	Unauthenticated requests reverted automatically by EVM execution.	Pass
Explanation and State Manipulation	Altering categorization logic or untracked state changes.	Dynamic user filtering via Filter aligned with real-time auditability in Track User / Track User contract.	Tampered local evaluation states failed verification against tracked blockchain state.	Pass

6.3 System Performance and Gas Consumption Analysis

To evaluate the operational efficiency of the architecture. The gas costs, transaction confirmation times, and system throughput were measured using the Remix IDE with an In-Memory EVM configuration. The evaluation also considered the balance between data privacy and blockchain storage requirements. Raw candidate data and XAI features were kept off-chain, while only non-invertible SHA-256 hashes were stored on the blockchain.

Quantitative execution performance is summarized in Table 3:

- (1) State Modifications (setUser function): Updating on-chain state and emitting immutable TransactionRecorded event logs required a total transaction cost of 27,983 gas (with an execution cost of 6,919 gas), achieving an average confirmation latency of 320 ms and a throughput of 48.5 TPS.
- (2) Read-Only Ledger Calls (trackUser / get user function): Non-transactional EVM view calls incur zero gas overhead when queried externally, returning verification state in 45 ms with throughput exceeding 200 TPS.
- (3) Internal Routing (fallback function): Inter-contract delegatecall routing logic required an internal gas overhead of 2,424 gas.

Table 3. Quantitative Smart Contract Execution, Latency, and Gas Overhead Analysis.

Operation / Smart Contract Function	Transaction Type	Execution Gas Cost (gas)	Total Gas Cost (gas)	Avg. Latency (ms)	Operational Throughput (TPS)
-------------------------------------	------------------	--------------------------	----------------------	-------------------	------------------------------

Deploy Dynamic Proxy Contract	Contract Deployment	425,100	685,420	1,450	—
Set user function	State Modification	6,919	27,983	320	48.5 TPS
Track User / get user function	External View Call	0	0	45	High (>200)
Fallback Function	Internal Delegatecall	2,424	2,424	—	—

6.4 Usability Assessment Methodology

Concurrently, non-functional testing was conducted to quantify the usability and interface quality of the autonomous framework. Although several usability evaluation tools were reviewed, the System Usability Scale (SUS) was selected as the primary metric. This selection was governed by the scale's robust reliability across small sample sizes, its high efficiency in differentiating interface accessibility, and its alignment with core software functionality benchmarks [27].

The study was carried out with several HRM professionals who worked with the system through the 15 test cases. At the end of the testing procedure, they answered the 10 questions of the SUS questionnaire, which had a scale from "strongly disagree" to "strongly agree." Five major parameters of usability were examined by the questions: structural complexity, ease of use, system integration, consistency, and confidence [28].

Given the specialized nature of the BC-XAIA framework, a purposive sampling strategy was employed to recruit expert evaluators capable of assessing both software interface accessibility and decision-explainability workflows. Table 4 outlines the demographic characteristics and technical background of the five evaluators.

Table 4. Evaluator Demographic Characteristics and Background.

Participant	Role / Profession	Domain Experience	Gender	Tech Familiarity
User 1	Senior HR Manager	8 Years	Female	Moderate
User 2	Technical Recruiter	4 Years	Male	High
User 3	HR Systems Specialist	6 Years	Female	High
User 4	Recruitment Specialist	5 Years	Male	Moderate
User 5	Recruitment Specialist	3 Years	Female	Low-Moderate

6.5 Usability Results and Comparative Analysis

The responses were computed using standard SUS scoring templates. The complete responses, raw scores, and finalized SUS percentage scores for all HRM participants are detailed in Table 5.

Table 5. BC-XAIA SUS Results.

Participant	1. I think that I would like to use this system frequently.	2. I found the system unnecessarily complex	3. I thought the system was easy to use.	4. I think that I would need the support of a technical person to be able to use this system.	5. I found the various functions in this system were well integrated.	6. I thought there was too much inconsistency in this system	7. I would imagine that most people would learn to use this system very quickly.	8. I found the system very cumbersome to use.	9. I felt very confident using the system	10. I needed to learn a lot of things before I could get going with this system.	SUS Raw Score	SUS Final Score
User 1	5	1	4	1	5	1	3	1	5	1	37	92.5

User 2	4	2	5	2	4	2	4	1	5	1	34	85
User 3	4	3	4	2	3	1	3	2	4	2	28	70
User 4	4	1	3	1	4	1	4	1	5	1	35	87.5
User 5	5	1	5	2	5	1	5	1	4	1	38	95
Average											34.4	86

As summarized from the data in Table 1, the system achieved individual participant usability scores of 92.5%, 85.0%, 70.0%, 87.5%, and 95.0%, respectively, which are graphically represented in the evaluation analysis matrix Fig. 7.

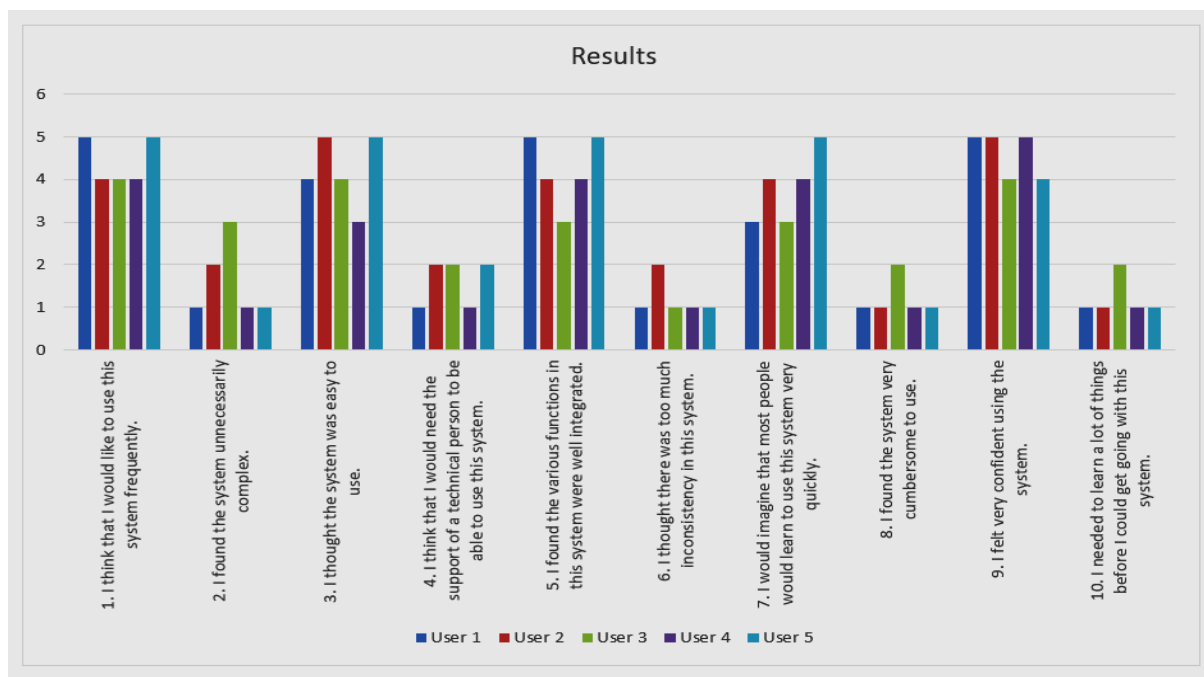


Figure 7. BC-XAIA Evaluation Results.

The individual participant responses yielded a mean SUS score of **86.0** ($SD = 9.78$, $Variance = 95.63$, $Median = 87.5$). To account for sample dispersion, a 95% Confidence Interval (CI) was calculated, yielding [73.86,98.14].

While the sample size ($n = 5$) represents a pilot evaluation limitation, usability literature in software engineering (e.g., Nielsen and Landauer) demonstrates that testing with 5 domain-knowledgeable evaluators is sufficient to identify up to 85% of core usability bottlenecks in early-stage application interfaces. Nevertheless, these preliminary findings serve as an initial proof-of-concept; future research will extend this evaluation to a larger sample size to conduct parametric significance testing across broader user demographics.

The SUS score across all participants was 86%, exceeding the standard usability benchmark of 68% [29]. The proposed BC-XAIA was benchmarked against relevant usability evaluations in decentralized software. As presented in Table 6, the BC-XAIA framework achieves higher usability scores compared to existing frameworks.

Table 6. Comparative Analysis of SUS Scores Across Different Frameworks and Benchmarks.

Study	Domain / Use case	Technology	SUS Score	Usability Rating
-------	-------------------	------------	-----------	------------------

Our framework (BC-XAIA)	Blockchain-based recruitment system	Blockchain, Smart Contract, XAI and Agile	86	Excellent
Awaji & Solaiman [30]	Academic credential verification	Ethereum and Smart Contracts	77.1	Good-Excellent
Bangor et al. [31]	General software	Various	68	Acceptable (baseline)

According to standard usability acceptability criteria represented in Table 6, an average score of 86% categorizes the BC-XAIA architecture within the "Excellent" and highly acceptable tier. Awaji and Solaiman [30] evaluated a decentralized credential verification system. And reported a score of 77.1, which falls within the "Good–Excellent" range.

7 Conclusion

This study provided BC-XAIA, an intelligent framework combining blockchain, smart contracts, XAI, and agile methodologies. It ensures reliable, transparent, and automated decision-making. The framework ensures data privacy, transparent decision-making, and reliable AI predictions. BC-XAIA was evaluated through 15 test cases with the participation of HR specialists. Explainability tools such as SHAP and LIME were used to ensure that predictions could be easily understood by users. Moreover, BC-XAIA obtained a usability score of 86% according to the System Usability Scale (SUS). Overall, BC-XAIA proves to be an effective, transparent, and secure solution enhancing security in multiple domains. Future work will focus on extending the system to larger datasets and conducting further real-world testing with more diverse user cohorts. Also refining the framework's explainability and usability features to support widespread adoption in HR systems and different domains.

DECLARATIONS

Funding. The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

Conflicts of interest. The authors declare that they have no competing interests.

Ethics approval. Participation in the usability evaluation was conducted in accordance with standard institutional ethical guidelines for non-invasive user feedback.

Informed consent. Voluntary informed consent was obtained from all individual participants prior to completing the usability evaluation; participation was entirely voluntary and anonymous. Consent for publication: not applicable.

Data and code availability. All data generated or analyzed during this study are included in this published article. The individual System Usability Scale (SUS) survey responses and calculated scores are fully provided within the article (Table 5). [The BC-XAIA implementation and code are available from the corresponding author upon reasonable request.]

Author contributions.

Hebat Allah Adel: wrote the main manuscript text and prepared figures and tables.

Hebat Allah Adel, Sayed AbdelGaber, and Wessam H. El-Behaidy reviewed the manuscript.

Use of artificial intelligence. No generative AI tools were used in data analysis, or coding for this manuscript.

REFERENCES

- [1] Calle-Zuniga, A., & Velasquez, W. (2026). A critical review on how cryptography and blockchain strengthen security in data networks. *Peer-to-Peer Networking and Applications*, 19(3), 86.
- [2] O. Kuznetsov, P. Sernani, L. Romeo, et al., On the integration of artificial intelligence and blockchain technology: A perspective on security, *IEEE Access*, vol. 12, pp. 3881–3897, 2024.
- [3] F. Bassan, M. Rabitti, From smart legal contracts to contracts on blockchain: an empirical investigation, *Computer Law & Security Review*, vol. 55, pp. 106035, 2024. DOI: 10.1016/j.clsr.2024.106035
- [4] A. Gucciardi, Trustless contract management: a study on the benefits of blockchain-based smart contracts, Dissertation, Politecnico di Torino (Italy), 2023.
- [5] H. Adel, M. ElBakary, K. ElDahshan, D. Salah, BC-HRM: A blockchain-based human resource management system utilizing smart contracts, In *International Conference on Deep Learning, Big Data and Blockchain*, Cham (Germany), 2021, PP. 91–105.

- [6] B. U. I. Khan, K. W. Goh, A. R. Khan, et al., Resource management and secure data exchange for mobile sensors using Ethereum blockchain, *Symmetry*, vol. 17, no. 1, pp. 61–75, 2025.
- [7] N. S. Kumar, S. Brinthakumari, N. Partheeban, Navigating the blockchain landscape with a comprehensive survey of smart contracts, Ethereum, and consensus mechanisms, In 4th International Conference on Data Engineering and Communication Systems (ICDECS), IEEE, March 2024, PP. 1–6.
- [8] D. Hariyani, P. Hariyani, S. Mishra, M. K. Sharma, A literature review on transformative impacts of blockchain technology on manufacturing management and industrial engineering practices, *Green Technologies and Sustainability*, pp. 100169, 2025. DOI: 10.1016/j.grets.2025.100169
- [9] Y. Liu, J. He, X. Li, et al., An overview of blockchain smart contract execution mechanism, *Journal of Industrial Information Integration*, pp. 100674, 2024. DOI: 10.1016/j.jii.2024.100674
- [10] W. Li, Z. Su, R. Li, K. Zhang, Y. Wang, Blockchain-based data security for artificial intelligence applications in 6G networks, *IEEE Network*, vol. 34, no. 6, pp. 31–37, 2020. DOI: 10.1109/MNET.011.2000195
- [11] A. M. Krishna, A. K. Tyagi, Intrusion detection in intelligent transportation systems and its applications using blockchain technology, In International Conference on Emerging Trends in Information Technology and Engineering (IC-ETITE), IEEE, 2020, PP. 1–8.
- [12] F. U. Muram, M. A. Javed, Drone-based risk management of autonomous systems using contracts and blockchain, In IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER), IEEE, 2021, PP. 679–688.
- [13] N. Kamble, R. Gala, R. Vijayaraghavan, E. Shukla, D. Patel, Using blockchain in autonomous vehicles, In Artificial Intelligence, Blockchain, and Future Cybersecurity Applications, Cham (Germany), 2021, PP. 285–305.
- [14] S. Jain, N. J. Ahuja, P. Srikanth, et al., Blockchain and autonomous vehicles: Recent advances and future directions, *IEEE Access*, vol. 9, pp. 130264–130328, 2021.
- [15] R. Gupta, A. Kumari, S. Tanwar, A taxonomy of blockchain envisioned edge-as-a-connected autonomous vehicles, *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 6, pp. e4009, 2021. DOI: 10.1002/ett.4009
- [16] P. Dakić, M. Živković, An overview of the challenges for developing software within the field of autonomous vehicles, In 7th Conference on Engineering of Computation Based Systems, 2021, PP. 1–10.
- [17] A. K. Tyagi, D. Agarwal, N. Sreenath, SecVT: Securing the Vehicles of Tomorrow using blockchain technology, In International Conference on Computing, Communication and Informatics (ICCCI), 2022, PP. 1–6.
- [18] M. Hataba, A. Sherif, M. Mahmoud, M. Abdallah, W. Alasmay, Security and privacy issues in autonomous vehicles: a layer-based survey, *IEEE Open Journal of the Communications Society*, vol. 3, pp. 811–829, 2022.
- [19] M. Parlak, Blockchain-based immutable evidence and decentralized loss adjustment for autonomous vehicle accidents in insurance, *arXiv preprint arXiv:2303.18130*, 2023.
- [20] S. K. Rana, A. K. Rana, S. K. Rana, et al., Decentralized model to protect digital evidence via smart contracts using layer 2 polygon blockchain, *IEEE Access*, vol. 11, pp. 12345–12355, 2023.
- [21] H. D. Zubaydi, P. Varga, S. Molnár, Leveraging blockchain technology for ensuring security and privacy aspects in Internet of Things: a systematic literature review, *Sensors*, vol. 23, no. 2, pp. 788, 2023. DOI: 10.3390/s23020788
- [22] J. Ahmad, M. U. Zia, I. H. Naqvi, et al., Machine learning and blockchain technologies for cybersecurity in connected vehicles, *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 14, no. 1, pp. e1515, 2024.
- [23] H. Guo, X. Yu, A survey on blockchain technology and its security, *Blockchain: Research and Applications*, vol. 3, no. 2, pp. 100067, 2022. DOI: 10.1016/j.bcr.2022.100067
- [24] A. Kharche, S. Badholia, R. K. Upadhyay, Implementation of blockchain technology in integrated IoT networks for constructing scalable ITS systems in India, *Blockchain: Research and Applications*, vol. 5, no. 1, pp. 100188, 2024. DOI: 10.1016/j.bcr.2024.100188
- [25] M. J. F. Iglesias, *Introduction to Blockchain, Smart Contracts and Decentralized Applications*, 1st ed. London (UK): Academic Press, 2023. ISBN 9781234567890
- [26] H. A. Adel, S. AbdelGaber, and W. H. El-Behaidy, "BC-XAIA: A Blockchain-Based Recruitment Framework with Explainable AI and Smart Contract Integration," *Applied Sciences*, vol. 16, no. 16, p. 8064, 2026. <https://doi.org/10.3390/app16168064>
- [27] usability.gov (2025) System Usability Scale (SUS). Available at: <https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html> (Accessed: 1 April 2025).
- [28] N. Thomas. How to use the System Usability Scale (SUS) to evaluate the usability of your website. 3 pages. [Online] Cited 2026-07-01. Available at: <https://usabilitygeek.com/how-to-use-the-system-usability-scale-sus-to-evaluate-the-usability-of-your-website/>
- [29] J. Blattgerste, J. Behrends, T. Pfeiffer, A web-based analysis toolkit for the System Usability Scale, In Proceedings of the 15th International Conference on Pervasive Technologies Related to Assistive Environments, 2022, PP. 237–246.

- [30] B. H. M. Awaji, E. Solaiman, Design, implementation, and evaluation of blockchain-based trusted achievement record system for students in higher education, In Proceedings of the 14th International Conference on Computer Supported Education (CSEDU 2022), vol. 2, PP. 225–237, 2022.
- [31] A. Bangor, P. T. Kortum, J. T. Miller, An empirical evaluation of the System Usability Scale, International Journal of Human–Computer Interaction, vol. 24, no. 6, pp. 574–594, 2008. DOI: 10.1080/10447310802205776.