IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 23 June 2024, accepted 8 July 2024, date of publication 12 July 2024, date of current version 23 July 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3426993

RESEARCH ARTICLE

Reputation Evaluation Using Fuzzy Logic for Blockchain-Based Access Control in an IoT Environment

ARWA A. ALQBAISHI[1] AND ALAA E. S. AHMED[1,2]
[1]College of Computer and Information Sciences, Imam Mohammad Ibn Saud Islamic University (IMSIU), Riyadh 11432, Saudi Arabia
[2]Shoubra Faculty of Engineering, Benha University, Cairo 13511, Egypt

Corresponding author: Arwa A. Alqbaishi (alqbaishi_arwa@hotmail.com)

This work was supported by the Deanship of Scientific Research at Imam Mohammad Ibn Saud Islamic University (IMSIU).

ABSTRACT To develop access control mechanisms, particularly in terms of maintaining effective and secure access control within Internet of Things (IoT) networks. Whereas the sufficient use must be made of blockchain-based access control technology. This is because of the sheer volume of connected devices and the subsequent increase in transactions. This can negatively impact the performance and responsiveness of the networks. Thus, this article proposes a comprehensive approach that evaluates the requester's reputation with respect to regulating access requests for IoT resources. The proposed approach combines fuzzy reputation with a decay algorithm. It then calculates the quantitative reputation value for each IoT user. This considers multiple variables, such as the Access Request Rate, frequency of requests, etc. This new reputation value serves as the basis for access-control decisions. Extensive simulations and experiments are conducted to evaluate the effectiveness of the proposed framework. For the simulation, we used a single-board Raspberry Pi. We also used a hybrid blockchain network environment comprising Geth and Hyperledger Fabric. We then analyzed and compared the performance of the proposed framework with that of the existing approaches. The results demonstrated that compared to the mathematical mechanism, the framework provides improved access control in IoT networks. The fuzzy-based reputation framework captures the dynamic nature of IoT environments, and effectively identifies trustworthy and malicious devices, whereas the decay algorithm reflects the most recent user behavior.

INDEX TERMS Access control, blockchain, decay, fuzzy logic, hyperledger FireFly, Internet of Things, smart contract.

I. INTRODUCTION

Recently, the number of interconnected devices in wireless networks has significantly increased. This has led to the formation of the Internet of Things (IoT), which generates and exchanges large amounts of data. However, the rapid growth of IoT devices has introduced multiple security and trust challenges [1]. Thus, to regulate access and data exchange within an IoT environment, establishing robust access control mechanisms and monitoring the behavior of entities are critical for ensuring the security and reliability of entity interactions. Blockchain technology has emerged as a promising solution to enhance access control. However, it offers decentralized and tamper-resistant access control mechanisms in IoT environments [2]. It improves security and trust among IoT devices, users, and service providers by leveraging the transparency and immutability of the blockchain. Furthermore, recent blockchain-based access control has heightened the need to monitor the behavior of access requests, in order to proactively block access requests from misbehaving IoT users and devices. Previous research indicates that various access behavior indicators significantly impact the performance and security of IoT [3]. Notably, blockchain-based access control is strengthened by different evaluation techniques that regulate access by evaluating the reputations of entities. This evaluation can assess the

The associate editor coordinating the review of this manuscript and approving it for publication was Yu-Da Lin.

trustworthiness and reliability of entities in IoT ecosystems, which is crucial in the decision-making processes and facilitates secure interactions within the IoT environment. This ensures that trustworthy entities are granted appropriate access privileges. Previous research has investigated various evaluation methods, the further discussion of which is presented in the literature section of this article. Prior studies have proposed several methods based on mathematical calculations for evaluating reputation values. Zhang et al. [4] proposed a mathematical evaluation method. The proposed framework provides static and dynamic access validation through an access control smart contract, and the static process validates access requests with policy lists. This dynamic process checks for concurrent access requests. A comprehensive reputation evaluation that considers the inactive state was proposed by Tu et al. [5]. The proposed model designs a Dynamic Evaluation Mechanism (DEM) that dynamically detects malicious behaviors. The DEM consists of two dynamic reputation evaluation algorithms: the Dynamic Evaluation Window Algorithm (DEWA), which adjusts the evaluation window dynamically, and the Reputation Hierarchical Decay Algorithm (RHDA), in which the reputation value is updated when IoT devices or users are inactive. In contrast, several studies have integrated a Fuzzy Inference System into reputation evaluation [6]. This serves to handle uncertainties and imprecise the data in an IoT environment. Fuzzy logic offers a flexible and intuitive framework for modeling and reasoning with vague or uncertain information. This bridges the gap between human-like reasoning and machine-based decision-making.

This is motivated by the recognition that blockchain-based access control mechanisms often struggle to capture the nuanced and dynamic nature of user reputation in IoT environments. The system proposed in this article is built on the framework provided in [4], which aims to improve the behavior validation process by employing the proposed Fuzzy Inference System (FIS). However, the design and implementation of fuzzy systems is complex. This is particularly true when the aim is to provide comprehensive reputation evaluation. The decay algorithms proposed in [5] were adopted to enhance the accuracy and responsiveness of reputation evaluations by considering the recency and relevance of user actions. Both articles are based on mathematical calculations; the former framework aims to detect and block concurrent access requests, whereas the latter provides a comprehensive evaluation framework with two decay algorithms, DEWA and RHDA.

This article contributes to the IoT field by developing a blockchain-based access control framework that can comprehensively evaluate reputation. Two periods were considered, active and inactive. Integrating decay algorithms into the process of evaluating reputation acknowledges the time related aspects of user behavior and reputation evolution. By considering the recentness and relevance of user actions, decay algorithms ensure that reputation values accurately reflect current and trustworthy information. This dynamic adjustment enhances the precision and responsiveness of reputation assessments, and provides better decision making regarding access control. In conjunction with this, the novelty of this paper lies in the integration of fuzzy logic and decay algorithms in blockchain-based access control, which can effectively handle the complexity and uncertainty associated with IoT environments. Subsequently, the proposed approach is implemented using a hybrid blockchain environment for the distributed implementation of the IoT network. The results of the evaluation analysis and comparative study reveal that the proposed framework can respond to access requests correctly and efficiently, eliminating the uncertainty contained in the mathematical calculations given by [5], and reducing the cost of the access requests deployed to the blockchain compared with that described in [4]. The main contributions of this article are as follows:

1) To address the recency and relevance of user actions by adopting the decay algorithms DEWA and RHDA [5].
2) To improve the behavior validation process [4], a Fuzzy Inference System (FIS) was integrated for use in reputation evaluation. Access requests are evaluated based on the user reputation in place of TimeOfUnBlock [4]. The FIS uses crisp inputs, such as the decay reputation that results from the employed decay algorithms, DEWA and RHDA.
3) To extend the capabilities of the previous framework, a hybrid blockchain environment was presented, where the framework in [4] was deployed to Ethereum and the reputation evaluation framework was deployed to the Hyperledger Fabric using Hyperledger FireFly.

By integrating blockchain-based access control, reputation evaluation, and fuzzy logic in IoT environments, this article advances existing knowledge on secure and trustworthy IoT systems. The structure of the article is as follows: In Section II, we provide an overview of the background to the study; Section III presents a literature review; and Section IV explores the proposed approaches to reputation evaluation for blockchain-based access control in IoT environments. Section V presents the framework settings needed before the evaluation. Section VI presents assessments and analyses of the results obtained during the evaluation of the proposed system. Section VII discusses some potential limitations to further refine and expand the capabilities of the proposed system. Finally, Section VIII presents the conclusions of the article.

II. BACKGROUND OVERVIEW

This section provides an overview of the background to the fields explored in this article, namely, trust, reputation, behavior, blockchain technology, blockchain-based access control, fuzzy theory, and last the decay algorithms. Understanding these concepts and their applications is crucial for the development of novel approaches to address the security, privacy, and efficiency challenges in computer science.

A. TRUST, REPUTATION, AND BEHAVIOR IN COMPUTER SCIENCE

A consensus on the definitions used in this field is required because of the divergence of the proposed evaluation methods [3], which are calculated using different measures. The concept of trust in computer science is derived from real life environments. It is not specific but depends on the context and purpose of its use. In general, trust is recognized as an entity that does not harm other entities based on previous experiences. Reputation refers to the overall history of an entity's behavior, where behavior refers to an entity's actions towards a resource. Such behavior can be classified as normal or abnormal, based on predefined conditions. Moreover, monitoring user interactions with IoT devices can help overcome the security threats caused by trust issues. The behavior monitoring process depends on several parameters that can be inferred experimentally or derived from previous research. The evaluation of user behavior in detecting malicious attacks involves various techniques. However, the proposed framework adopts the algorithms proposed by the authors [5] in another framework [4]. Therefore, it is crucial to identify and redefine the three terms based on the two frameworks and the proposed framework processes, as follows:

Trust: The model presented in [5] expresses trust as the interval used for reputation evaluation, where a long interval means that the user has a good reputation and is trusted; otherwise, the user is not trusted. The authors of [4] stated that whenever a user satisfies the policies and there is no successive access request, they are considered trustworthy. The proposed framework combines the concepts of the two systems [4], [5], and trust is expressed when policies are satisfied. The interval used for reputation evaluation prevents successive access requests.

Reputation: In the model presented in [5], reputation is used to express trust. Direct and indirect reputation evaluations are considered to achieve comprehensive reputation evaluation. For a direct reputation, the behavior of IoT users/devices and two mathematical functions are used. The first mathematical function is used only when all user behaviors are normal, because it cannot resist discrimination attacks. However, the second function, which is an improved version of the first mathematical function, can resist such attacks and is used once the abnormal behavior has been evaluated. Indirect reputation assessment considers the access gateway's reputation and the users' access through the gateway. In [4], reputation was not expressed by any means. Because the algorithms presented in [5] are adopted in the proposed framework, a different reputation concept results in a reputation being inferred after applying fuzzy logic.

Behavior: The model presented in [5], which combines positive, negative, and penalties, was calculated using a mathematical function. If the result is below the normal threshold, misbehavior is present. As described in [4], when concurrent access requests do not exceed the minimum allowable time, misbehavior is present. Moreover, behavior evaluation consists of two steps, the combination of which infers behavior status. A variable called the Time to Last Request (ToLR) is compared to the minimum allowable time interval parameter. If the period between two successive requests is less than or equal to the minimum allowable time, the second request is considered a frequent request. In the second step, if the variable Number of Frequent Requests (NoFR) exceeds the threshold parameter, a misbehavior is detected. The proposed framework adopts the same concept as [4], where behavior denotes the calculation of both positive and negative access behaviors.

B. BLOCKCHAIN-BASED ACCESS CONTROL

The integration of blockchain and access control provides an innovative approach to the management and security of access to digital resources and systems. Blockchain-based access control aims to provide a decentralized and transparent solution. It leverages the characteristics of blockchain technology. These include immutability, transparency, and distributed consensus. In additional, it overcomes the issue of controlling IoT access requests and adapts to the heterogeneous and dynamic nature of the IoT environment by providing a lightweight, scalable, and trustworthy access control system. The exploitation of blockchain properties provides decentralized policy management for the storage of access control policies and access decision processes.

Furthermore, in a blockchain-based access control system, the access permissions and identity information are stored in a tamper-resistant decentralized ledger. Each user or entity is assigned a unique cryptographic identity, usually as a digital wallet or a public-private key pair. Several attempts have been made to incorporate blockchain technology into access control systems. One proposed system is fabric-IoT, developed by Liu et al. [7], which uses a smart contract to deploy the Attribute-based Access control (ABAC) model and store policies on the Hyperledger Fabric blockchain, thus ensuring the integrity of the process. This system eliminates the need for trusted third parties and centralized processing, thus providing scalable, fine-grained access control management. Blockchain was also used in [8] and [9] to store and validate access-control policies. However, the computational capabilities of blockchains have only recently been considered for processing evaluations of access request behaviors.

C. FUZZY THEORY

Fuzzy logic addresses the uncertainty and imprecision in data [3], [10]. A fuzzy logic system consists of three main components: fuzzification, Rule-Based Inference, and defuzzification. Fuzzification maps crisp numerical values to linguistic terms such as ''high'', ''low'', and ''medium''. The rule-based inference is then expressed in an ''if–then'' format, which combines the input fuzzy sets according to

the rules to generate fuzzy output sets. In the final step, defuzzification converts the fuzzy output sets back to crisp values. The leveraging of fuzzy sets and linguistic rules provides a flexible and interpretable approach for modeling complex systems.

D. DECAY ALGORITHMS

The algorithms employed, DEWA and RHDA, were obtained from [5]. The algorithms state that the DEWA updates the window value whenever there is a list of behaviors. Otherwise, RHDA is called to update the reputation value during the user's inactive state. Therefore, Algorithm 1 preserves the influence of these two algorithms. DEWA [5] aims to update the window value based on the reputation. If reputation is within the normal threshold of 0.5, it will increase throughout the three stages. The first stage is the slow start stage, where reputation increases exponentially. The second stage is a fast increase, in which reputation grows linearly. The third stage is the keep steady stage, in which reputation is set to a window threshold value. However, if the reputation is below the normal threshold, the window value moves to the rapid reduction stage. The last stage showed at slow decrease. When the reputation is below the normal threshold and the user is inactive, reputation gradually decreases. The RHDA algorithm [5] was introduced for inactive scenarios by calculating the time difference between the current time and LastActiveTime. If the user is inactive for a short period of time, then the reputation value remains unchanged. If the user exceeds a short period but not a long period, the reputation will decrease linearly and will not decay too quickly. Finally, the reputation value decreases exponentially if the user exceeds a long period of inactivity.

III. LITERATURE REVIEW

This section reviews the literature on the use of blockchain-based access control mechanisms for reputation evaluation in IoT environments. Liu et al. [11] proposed a trust management system taxonomy and classified evaluation methods based on whether trust values must be computed. This classification was divided into three categories. The first category is trust-value-based, which uses specific trust values calculated using a trust model. The second category is trust-value-free and does not rely on numerical trust values to evaluate trust relationships between entities. The last category is a hybrid of the first two. Liu et al. [11] focused on the first and second subcategories of the trust-value-based category, namely inference models and weighted average models. The following subsections discuss different approaches proposed for each category.

A. INFERENCE MODELS (IMS)

The models described in this section use a significant amount of trustworthy evidence to quantitatively describe trust through inferences. For instance, probability-based IMs use discrete trust evidence to calculate trust values and facilitate trust reasoning. This is achieved by introducing probability distributions or density functions, along with the likelihood of events occurring. Building upon the utilization of fuzzy logic, Zulkifl et al. [12] introduced a novel framework called Fuzzy and Blockchain-Based Adaptive Security for Healthcare IoTs (FBASHI), which is based on the use of fuzzy logic and blockchain to provide authentication, authorization, and audit log functions of IoT network security. It comprises of three FISs: authentication, trust evaluation, and access control. The proposed framework successfully achieves distributed trust based on fuzziness, and removes single points of failure from the IoT network.

Other studies combined fuzzy logic with various methods to enhance the performance of fuzzy systems. For instance, Esposito et al. [13] proposed a decentralized trust-management mechanism derived from game theory for dynamic access control based on the ABAC model. The aim was to achieve robust, decentralized trust management that could tolerate the transmission of false values by malicious nodes. The authors modeled the interaction of the IoT and edge nodes according to a Bayesian Signaling Game (BSG) and developed mechanisms to exclude nodes suspected of being malicious. In addition, trust calculation employs fuzzy theory, which uses linguistic terms and fuzzy sets to handle the uncertainty in reputation scores. These scores are then aggregated using the Dempster–Shafer (DS) theory and pignistic probability transformation. In [14], the authors proposed an adaptive and dynamic risk-based access control model that utilizes real-time and contextual information to determine access decisions. The proposed model combines the Fuzzy Inference System with expert judgment to provide consistent and realistic risk values for various access control operations. Furthermore, smart contracts are employed to track and monitor user activities during access sessions in order to detect and prevent potential security violations.

A secure framework was proposed by Gardas et al. [15] to integrating edge and blockchain technologies into IoT networks to ensure data protection and energy efficiency. This provides a platform for node selection in various IoT–edge frameworks. The model design involves developing a novel node selection approach for blockchain-enabled edge IoT, which utilizes a fuzzy-based technique and the Fuzzy Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) approach to rank IoT node alternatives. Wang et al. [16] designed Trust Management Models (TMs) to implement a lightweight ABAC framework for blockchain-enabled IoT. All historical access requests and authorization results are recorded using a PoA-based blockchain. The system then used these records to construct a graph. Next, a Markov random walk is performed on the model to compute its trustworthiness, which measures the probability of trustworthiness existing in the target device. Alternatively, the ambiguity of trust arises because it is not a binary judgment, but rather a fuzzy state between trust and distrust that is difficult to quantify precisely. To address this issue,

a model based on fuzzy theory, specifically fuzzy logic, can deal with imprecision and uncertainty and allow for a quantitative analysis of trust ambiguity.

Yuan et al. [17], proposed a novel system called Street Engine, which was introduced as a new low-power computing chip powered by a customized rule fuzzy system referred to as Street Language. To demonstrate the feasibility of this approach, the authors simulated the homing behavior of honeybees, demonstrating how a street agent-based system can achieve simple cognitive behavior by integrating multiple hard-wired natural reactions, similar to insect neural networks. For autonomous vehicles, Lian et al. [18] proposed a novel fuzzy-based system to address the problem of nonlinear networked autonomous vehicle systems under multiple cyber-attacks. The authors mitigated the network burden imposed by the control network by proposing an asynchronous resilient event-triggered scheme (ETS), and developed a dynamic output-feedback control method to handle the lateral control problem for networked autonomous vehicles. Yazdinejad et al. [19] designed a novel, secure, and intelligent fuzzy blockchain framework for network attack detection driven by uncertainty issues associated with deep learning techniques. The framework incorporates a fuzzy deep-learning model, fuzzy control systems, and fuzzy matching modules for network attack detection. The authors compared the results with those of the fuzzy classifiers. The framework uses metaheuristic algorithms for optimization and conducts fuzzy matching to detect fraud. The evaluation results demonstrated the efficiency and effectiveness of the framework in detecting threats and making decisions in IoT networks based on blockchain technology.

B. WEIGHTED AVERAGE MODELS

Several attempts have been made to express trust through a continuous score, automate reputation aggregation and revelation, and ensure robustness against blockchain attacks and trust evaluations. Various studies have investigated the application of blockchain technology in managing IoT workflows, to ensure their integrity, trust, and reliability. Zhang et al. [4] proposed a smart-contract-based access control framework to overcome security threats related to centralization. Three smart contracts are deployed in the blockchain for various purposes: Access Control Contracts (ACCs), Judge Contract (JC), and Register Contract (RC). This provides distributed and trustworthy access control in the IoT. The proposed framework provides static and dynamic access validation through ACC, where the static validation process searches for policy lists. The dynamic validation process checks for any misbehaving access activity and then informs the JC, which then judges and returns the penalty. The experimental results prove the efficiency of the framework in detecting and preventing malicious acts.

Inspired by the previous model [4], Lone and Naaz [20] proposed a Reputation driven dynamic Access Control (Rep-ACM) framework that integrates a permissioned blockchain and smart contract capabilities with a reputation service to build the reputation of the subject. This framework integrates smart contracts for policy enforcement and reputation services to penalize subjects based on their behavior with object resources. The results indicate that the proposed Rep-ACM framework reduces the number of smart contracts to be deployed compared to the number required in the scheme proposed by Zhang et al. [4]. They also identified and addressed a potential Denial-of-Service (DoS) attack vulnerability that arose in Zhang et al.'s study [4]. Additionally, the proposed framework achieved a higher number of Transactions Per Second (TPS) than Zhang et al. [4].

In the Industrial IoT (IIoT), Wu and Ansari [21] designed a new voting mechanism that includes a trust evaluation for access control in blockchain-based IIoT Groups (IIoT-G) that prevents malicious devices from spreading misleading information. The voting results were determined based on trust values and feedback using an equal-weight voting mechanism. In this system, blockchain is not directly employed for trust management; it is deployed in IIoT Devices (IIoT-Ds) to ensure access control. Based on trust evaluation, different devices are assigned varying weights during authorization voting, thereby ensuring that the data can be trusted. Song et al. [22] proposed a distributed IoT security system architecture based on blockchain intended to proactively detect various network threats and quickly respond to any malicious behavior. The proposed system technologies consist of three main modules. The access control module adopts the ABAC model for authorization and decision-making processes. The separation mapping module intercepts any unauthorized incoming packets that reach the destination nodes through the core network. Finally, the security feedback module monitors the access requests and informs other modules to initiate a quick response to block malicious users. It identifies abnormal traffic by monitoring and analyzing traffic behavior using a detection method based on statistics [23], locating the identity and behavior record of an abnormal user, and then evaluating the detected user traffic to avoid false alarms.

A comprehensive reputation evaluation that considers the inactive state was proposed by Tu et al. [5]. They developed the Blockchain-based Trust and Reputation Model (BTRM), which combines blockchain with Trust and Reputation Management (TRM). The developed model overcomes emerging issues, such as the use of fixed intervals in the evaluation process, which makes it challenging to detect attackers with intelligence. Designing a DEM that dynamically detects malicious behavior. The DEM consists of two dynamic reputation evaluation algorithms: the DEWA, where the BTRM model can adjust the evaluation window dynamically, and the RHDA, where the reputation value is updated by the BTRM model when IoT devices or users are inactive. Another issue that this model aims to address is the assessment of a single behavior causing an inaccurate reputation value, the proposed model provides a comprehensive user behavior assessment from three perspectives: link behavior, access

behavior, and communication behavior. An adaptive trust evaluation model for IoT called AITTE was proposed by Jiang et al. [24]. The model includes a multi-level smart contract system that determines the authority level based on device attributes and trust degrees, allowing for dynamic authority allocation. The model also employs several trust evaluation algorithms, including the Historical Trust Calculation Algorithm, the Recommended Trust Calculation Algorithm, and the Comprehensive Trust Value Calculation Algorithm. Adaptive fusion weights were introduced to reduce the influence of malicious recommendation behavior and effectively improve the evaluation accuracy of trustworthiness. Yang et al. [25] proposed a blockchain-empowered Token-Based Access Control (TBAC) system that includes user reputation evaluation to enhance data security and privacy protection. The access control process is divided into three stages: policy upload, token request, and resource request. The system uses smart contracts in a blockchain to store and process user and reputation information, access control policy information, and token information. It also incorporates a user reputation evaluation module to provide feedback on the access control. The system addresses issues such as coarse-grained access control, poor manageability, and security threats in the existing access control systems. For rural management, Arsyad et al. [26], proposed a farm transaction model using the Encapsulating Block Mesh (EBM) platform, integrating blockchain technology for secure farm operations and employing NFC tags for information transfer and data storage. The method involved a unique blockchain design and the ''bucket principle'' for farm transactions, ensuring data integrity and traceability. Trust evaluation was conducted by analyzing the secure documentation of farm transaction records using the EBM platform. The method tested user trust by validating the chain of events in the blockchain system, encapsulating and linking farm transaction blocks to ensure data integrity and traceability. The authors used the case of cocoa production, and the results demonstrated the feasibility and effectiveness of the approach.

Sun [27], integrated blockchain technology into rural financial management to enhance security and transparency, with the use of embedded systems for data acquisition and processing. The system uses a secures interface that regulates access, allowing only authenticated users to access and transmit data to the blockchain platform. To evaluate the system's performance, this study employed a combination of quantitative and qualitative techniques, such as a questionnaire survey, in-depth user interviews, and an online feedback portal, enabling continuous monitoring and collection of user opinions and attitudes. The paper also proposed mathematical formulas to evaluate operational efficiency, consensus mechanism probability, and data security incidents within the rural financial management cloud platform. An innovative solution facilitates the design of a trust evaluation framework using TRM, and some related work has investigated the implementation of TRM. Malik et al. [28] proposed a trust management framework to address the trust problem in blockchain-based supply chain applications. This framework uses a consortium blockchain to track interactions and assigns trust and reputation scores using TRM. The blockchain trust and credibility module assesses the consistency of goods and trust among participating organizations by observing the data layer. This is achieved through an automated procedure that uses intelligent contracts and blockchain for each transaction. An Access Control List (ACL) is used in the blockchain layer to ensure that the rule is fulfilled during the read-and-write data operation on the blockchain. In [29], the authors proposed a TRM model with the advantages of recursion and bidirectional interaction calculations between two nodes. The aim of the model was to solve IoT device limitations affecting authorization by deploying the ABAC model with TRM in the blockchain. The calculations in the TRM contract consist of two methods. First, the trust score calculations of provider nodes towards requesting nodes are based on their history of interactions; if no history exists, zero is given as the initial value. The results of the trust score calculations were either positive or negative. Second, reputation calculations were performed using the Gompertz function by aggregating the trust score values of the requesting node across different providers. In addition, a feedback mechanism is proposed, in which the requester's trust score and reputation are updated during the authorization process and data access. Putra et al. [30] developed a blockchain-based TRM for IoT access control using the ABAC model, which progressively evaluates and calculates trust and reputation scores of each participating node to achieve a self-adaptive and trustworthy access control system. The authors adopted the Gompertz function for trust value calculation and formulated a reputation score for global trust computation. Progressive trust and reputation evaluation may help effectively detect and eliminate malicious or compromised nodes in the network.

Overall, the reviewed literature strives to contribute to the existing body of knowledge, bridging gaps in understanding and offering valuable insights into the research topic. The literature review highlights several benefits of the fuzzy reputation evaluation. First, it enhances the accuracy of access control by evaluating the trustworthiness of entities considering diverse contextual factors and dynamic changes in the IoT environment. Second, fuzzy reputation evaluation improves overall security by enabling more fine-grained access control decisions based on reputation value. This helps to mitigate the risks associated with unauthorized access, malicious activities, and the presence of compromised devices or users in IoT systems. For instance, Zulkifl et al. [12] proposed an evaluation method based on fuzzy theory.

Despite these positive findings, it is essential to acknowledge the limitations and challenges identified in the literature. Some studies have highlighted the complexity of designing and implementing fuzzy reputation evaluation systems,

including the need for appropriate membership functions, fuzzy inference mechanisms, and aggregation methods. Furthermore, scalability, privacy protection, and potential for reputation manipulation have been discussed as potential challenges. Several studies have driven further development and combined fuzzy logic with other techniques such as Expert Judgment [14], TOPSIS [15], game theory [13], Markov random walk [16], ETS [18], probabilistic reasoning [17] and deep learning [19]. In contrast, other researchers have employed different evaluation methods based on mathematical calculations and probabilities [20] and [30]. To implement a comprehensive evaluation framework that considers the users' inactive state, prior research proposed a decay algorithm [5], [24], [28], and [30], to influence the reputation value during users' inactive state where all of these methods are based on mathematical evaluation approaches. To implement fuzzy-based reputation evaluation, a substantial number of primary studies (Zulkifl et al. [12], Gardas et al. [15], Esposito et al. [13]) have used the Hyperledger Fabric platform. In contrast (Atlam et al. [14]) used the Bitcoin platform to implement fuzzy logic methods for blockchain-driven access control mechanisms in IoT.

In conclusion, the findings from this literature review demonstrate that fuzzy reputation evaluation can significantly enhance access control accuracy, improve overall security, and foster trust in blockchain-based access control systems within the IoT domain. However, little attention has been devoted to providing secure and comprehensive trust management in the IoT. In this paper, we improved the process of access control decision [4] by integrating fuzzy logic and employing decay algorithms [5], utilizing the Hyperledger Fabric platform to implement a fuzzy-based reputation evaluation. Addressing these research gaps will further leverage the potential of fuzzy reputation evaluation to enhance access control and security in the evolving IoT of landscape.

IV. METHODOLOGY

This section presents the system design, in which a mixed methods approach is adopted to achieve the required objective. The proposed access control method combines and integrates a Fuzzy Inference System for reputation values and two algorithms, RHDA and DEWA [5], into the proposed access control method drawn from [4]. The reputation evaluation process based on the blockchain is presented in the following subsections. First, the architectural design of the system was explained. Second, a Hybrid blockchain environment is presented. Third, we describe the parameters and factors employed in the proposed framework. Fourth, we presented the proposed algorithms for reputation evaluation. Finally, we describe the design of Fuzzy Inference Systems.

A. ARCHITECTURE OF THE SYSTEM

The overall structure of the system consisted of a Geth network with four devices. Table 1 and Fig. 1 present the devices used to build the Geth network, which was created in the same way as the system presented in [4] by employing the same abilities as Raspberry Pi, but with the capabilities of different computers. Further, a Bootnode Geth node PC1 acts as a miner and hosts the HyperLedger Fabric chaincode deployed to FireFly. This node also serves as an entry point for other nodes; it plays a crucial role in facilitating network connectivity and peer discovery, ensuring that the decentralized nature of the blockchain network is preserved and promotes a robust and distributed network infrastructure. PC2 is also a miner, and connects to bootnode. The other two Raspberry Pi nodes were connected to PC1 as light nodes. Raspberry Pi acts as an access gateway, where one of the devices acts as an object that connects to the remix with the aim of deploying Solidity smart contracts, adding policies, registering to the lookup table, and monitoring incoming access requests. The other Raspberry Pis acts as a subject that requests access and connects to the FireFly network to create an account using the public Geth key.

TABLE 1. System device specifications.

Device	Specifications
MacBook Pro	CPU: Apple M1, Operating System: macOS Ventura (13.3.1), Memory: 16 GB, Hard Disk: 1 TB
MacBook Pro	CPU:Intel, Operating System:MacOS, Memory:4 GB, Hard Disk: 64 GB
Raspberry Pi 3 Model B	CPU: Quad-core Cortex A53, 1.2 GHz, Operating System: Ubuntu server 22.04.2 LTS (64 bit), Memory: 1 GB RAM, Hard Disk: 64 GB (microSD card)
Raspberry Pi 3 Model B	CPU: Quad-core Cortex A53, 1.2 GHz, Operating System: Ubuntu server 22.04.2 LTS (64 bit), Memory: 1 GB RAM, Hard Disk: 64 GB (microSD card)

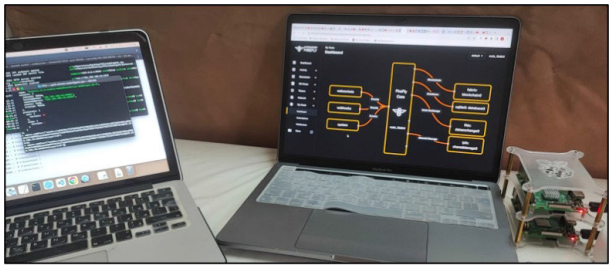


FIGURE 1. System devices consist of two laptops and two raspberry Pis. One laptop presents the FireFly stack whereas the other presents the mining process for the Geth network.

B. BLOCKCHAIN PLATFORMS

The proposed framework consists of a hybrid blockchain environment, in which an access control system is deployed to Ethereum [4], and the proposed reputation evaluation framework, which is implemented as Hyperledger Fabric chaincode, is deployed in the Hyperledger FireFly.

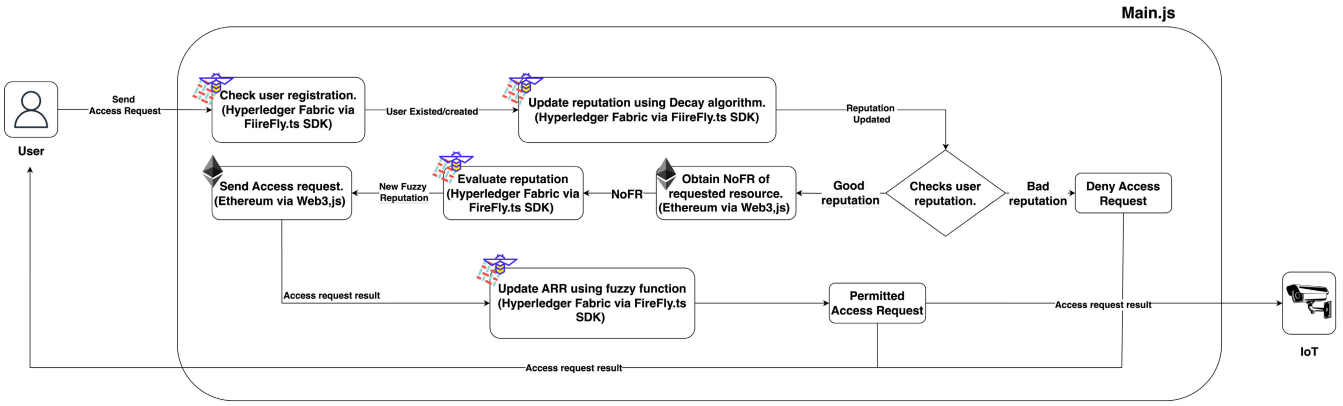


FIGURE 2. Illustration of the access control, the network consists of two blockchain platforms, the user requesting an access to resource and the Main.js responsible for processing the request and communicating with regarding blockchain platform.

The main reason for deploying Hyperledger Fabric is the limited capabilities of Solidity, as the use of fixed-point numbers has been proposed as a critical issue for calculating the reputation value. The system [4] calculates the penalty and variable TimeOfUnBlock, but Solidity discards the points, that are critical for the reputation value. Another reason is the support for fuzzy logic, as a substantial number of studies reviewed in the previous section (Zulkifl et al. [12], Gardas et al. [15], Esposito et al. [13]) used the Hyperledger Fabric platform. In contrast, Atlam et al. [14]) used the Bitcoin platform to implement fuzzy logic methods for blockchain-driven access control mechanisms in IoT. The researchers exploited customized blockchain platforms with different consensus algorithms, such as Reliable, Replicated, Redundant, And Fault-Tolerant (RAFT), Practical Byzantine Fault Tolerance (PBFT), and Proof of authority (PoA).

Furthermore, to initiate the connection and exchange of data between the two platforms, Hyperledger FireFly is used to deploy fabric chaincode [31]. FireFly aims to simplify the process for developers, and can be used to build and deploy DApps. It provides a scalable and interoperable solution for the management and integration of DLTs across different blockchain networks, but does not allow interoperability between blockchains, rather than fostering interoperability for the application tier. One of the key strengths of FireFly is its ability to connect multiple DLT platforms, thus enabling enterprises to leverage the benefits of various blockchain technologies simultaneously. It supports popular DLTs such as Ethereum, Hyperledger Fabric, and others, allowing developers and organizations to integrate existing blockchain networks or create new ones, as needed. However, a crucial configuration is required before starting the FireFly stack: updating the version of the fabric peer to the 2.4 version, the same as that used by the nodeJS by overwriting the FireFly docker file. After FireFly has successfully started and the chaincode has been packaged [31], the latter can be deployed in FireFly.

C. PARAMETERS AND FACTORS

The construction of the proposed framework is based on the system presented in [4] and algorithms presented in [5]. Hence, it is crucial to identify parameters and factors other than the new parameters and factors employed in the proposed framework. Table 2 presents the descriptions of the overall system parameters. These include the new parameters and factors utilized in the proposed system, such as fuzzy reputation, decay reputation, time-period variables (IdleTime, FirstTimeRequest, LastActiveTime, and StartSessionTime), PolicyResult, Geth ID, and ReputationList.

D. PROPOSED REPUTATION EVALUATION ALGORITHMS

The algorithms proposed for this system are presented in this section. Algorithm 1 demonstrates the process of reputation-based access control assessments. Algorithm 3 discusses the updates made to the original AccessControl ABI [4]. Algorithm 2 presents the fuzzy evaluation process used to determine the access request rate. An illustration of the proposed access control is presented in Fig. 2, to facilitate understanding and communication of the proposed system. The Main.js module is responsible for processing requests when a user sends an access request. First, it checks user registration in the system using a Hyperledger Fabric via the FifrFly.js module. Then, it updates the users' reputation values using decay algorithms. After updating the reputation for the user's inactive state, it retrieves the NoFR from Ethereum via Web3. It then sends the NoFR to the Hyperledger Fabric using the FifrFly.js module, to evaluate the user's reputation using a fuzzy function. Next, Main.js sends the access request to Ethereum along with the resulting fuzzy reputation. After evaluating the access request, Main.js updates the access request rate by using another fuzzy function. Finally, the user receives the results from their access requests. The following subsections introduce and explain the proposed algorithm.

TABLE 2. Main parameters employed in the proposed system.

Parameters	Description
σ	The decay factor used in the RHDA.
n_1	Constant for decay strategies used in the RHDA.
n_2	Constant for decay strategies used in the RHDA.
W_{init}	The initial evaluation windows.
n_{init}	The window factor used in the DEWA.
W_{th}	The threshold of the max evaluation window
W_f	The threshold window value for the fast increase stage
minInterval	The minimum allowable time interval between two successive requests.
Threshold	Threshold for the NoFR.
JC Base	A parameter that determines how the penalty changes with the length of the misbehavior record.
JC interval	A parameter that determines how the penalty changes with the length of the misbehavior record.
UserID	User ID are stored in the ledger state, initialized by the Geth PU key.
Decay_Rep	The reputation parameter will be updated by the RHDA.
Fuzzy_Rep	The reputation parameter will be updated by the fuzzy function.
Rep_list	A list to store the reputation values during an active session, which is emptied when the session ends. This parameter is used in the RHDA and DEWA algorithms.
policyResult	A list that stores the results of access requests. It is used to evaluate the ARR.
AccessRequestsRate	The result of the ARR fuzzy function, which is used as the input for the reputation function.
Last Active Time (LAT)	The time at which the user ends the access request in milliseconds.
Start Session Time (SST)	The time at which the user starts the access request in milliseconds.
First Time Request (FTR)	The time at which the user first requests access in milliseconds.
Current Time (CT)	The current time of the evaluation in milliseconds.
IdleTime	This parameter calculates the inactive periods.
NOFR	This parameter is updated by Geth after processing an access request and when frequent access has been detected.
Penalty	This parameter is updated by Geth after processing an access request and when the penalty function has been called.

1) REPUTATION-BASED ACCESS CONTROL ALGORITHM

Algorithm 1 is implemented in a JavaScript file and combines different functionalities implemented from both Fabric and Solidity perspectives to comprehensively evaluate access requests based on reputation. The algorithm requires the following variables: Resource, Action, Rep_{Fuzzy}, Current Time, and UserId. The first step is to check whether UserId already exists in the Fabric ledger state. If it does not, an error is prompted and the session ends. When UserId is registered in Fabric, the second step is to call the Fabric function to prompt time variables to start the session and then update the Rep_{Decay} during the inactive state of the user by calling the RHDA function. Three different time variables are initialized as discussed earlier for ARR-FIS the StartSessionTime, FirstTimeRequest, and IdleTime, and are used for the time period calculations in the Access Request Rate (ARR) algorithm. StartSessionTime is initiated at the current time, and FirstTimeRequest is assigned to the current time only if this is the first request, which can be achieved by checking the list of access request results. If it is empty, this is the first request made by the user. Finally, IdleTime calculates inactive time by checking LastActiveTime. IdleTime is calculated if it is not initialized to zero. Otherwise, IdleTime is initiated at zero because this is the user's first session. Following this, Line 5 in the algorithm obtains the user's information and updates the variables.

In Line 6, the entire loop prompts the user to request access and read its choice. In the first case, when the user requests access from lines 7 to 20, a Solidity function is called by web3 to obtain NoFR, following which a Fabric function is called to update the Rep_{Fuzzy}. In Line 12, an access request is sent by a Solidity function, and the access result is obtained. ARR is updated after every access request using Algorithm 2. In Line 16, the variables LastActiveTime, NoFR, and Window are updated by calling the DEWA algorithm. Before permitting the user to make another access request, Line 18 provides a proactive step by detecting the resulting Rep_{Fuzzy} value; if it is below the normal threshold of 0.5, the session ends and the system exits the loop with a bad reputation error. Conversely, in the second case, the session ends and the system exits the loop when the user no longer sends an access request. The end of the session is a Fabric function, that empties the list of reputation values and sets StartSessionTime to zero and LastActiveTime to the current time.

2) ACCESSCONTROL ABI ALGORITHM

Notably, there have been changes to the AccessControl ABI Algorithm [4]—the updated ABI is presented in Algorithm 3. However, the former AccessControl process flow is not affected, as only a reputation check replaces the TimeOfUnBlock in the latter. Hence, the TimeOfUnBlock check is used only to initialize the NoFR after penalty detection. In the AccessControl ABI Algorithm [4] process flow, the system first checks whether the user is in the block state by comparing its TimeOfUnBlock with the current time; if it is lower, the following variables are set to zero: NoFR, ToLR, and TimeOfUnBlock. Otherwise, access request is denied. The access request passes through both static and dynamic steps when the user is not blocked. The static step checks the policy, whereas the dynamic step checks the NoFR to detect the concurrent requests. The dynamic step sets NoFR to zero when the time between concurrent

Algorithm 1 Proposed Access Control Algorithm

Inputs: Resource, Action, Rep_{Fuzzy}, Current Time, UserId.
Outputs: Reputation, Time, NoFR, Access Results, Gas Used.

if userId is not registered in Fabric, **then**
  return Error user is not registered.
end if
Fabric Function: startSession(userId);
Fabric Function: FetchUserInfo(userId);
while true **do**
  if user is requesting access, **then**
    Solidity Function:
    $NoFR \leftarrow GetNoFR(MethodName, ResourceName, Action)$;
    Fabric Function: $Rep_{Fuzzy} \leftarrow fuzzy(NoFR, userId)$
    Fabric Function: $Rep_{Fuzzy} \leftarrow FetchUserInfo(acc_str)$
    Solidity Function:
    $AccessResult \leftarrow AccessControl(MethodName, Resource, CurrentTime, Action, Rep_{Fuzzy})$
    {Recalculate the ARR and update other parameters.}
    Fabric Function:
    EvaluteAccessRate(AccessResult,userId).
    Fabric Function: Updater(userId,LAT,NoFR).
    if $Rep_{Fuzzy} < 0.5$ **then**
      {End Session due to user's bad reputation.}
      Fabric Function: CloseSession(userId);
      Exit Loop
    end if
  else
    Fabric Function: CloseSession(userId);
    Exit Loop
  end if
end while

Algorithm 2 Fuzzy Access Request Rate (Fuzzy-ARR)

Inputs: Access Result, UserID
Outputs: ARR_{fuzzy}.
Require: StartSession, LastActiveTime, FirstReqTime, IdleTime, CurrentTime, $PermittedRequest \leftarrow 0$, $DeniedRequest \leftarrow 0$.

if UserID does not exist in the chaincode state **then**
  UserID does not exist.
else
  $user \leftarrow ObtainUserInfoFromChaincodestate$
  $user.policyResult \leftarrow addAccessResult$
  $ResultList \leftarrow user.policyResult$
  for $index \leftarrow 0$ `to ResultList.length` **do**
    if ResultList[index] is Permitted **then**
      $PermittedRequest++$;
    else
      $DeniedRequest++$;
    end if
    ActiveTime = (CurrentTime − StartSessionTime)
    $OldTime \leftarrow FirstReqTime + LastActiveTime$
    TimePeriod = (ActiveTime + (OldTime) − IdleTime)
    $ARR_{fuzzy} \leftarrow Fuzzy(TimePeriod, Permitted, Denied)$
  end for
end if

requests exceeds the minimum interval. The updated ABI in the Algorithm 3 access process flow first checks whether the user's reputation is within the normal threshold of 0.5. If this is true, the access request passes through the same static and dynamic steps. Conversely, if the user's reputation is below the normal threshold, access request is denied. Furthermore, if the TimeOfUnBlock value was less than the current time, the following variables were set to zero: NoFR, ToLR, and TimeOfUnBlock.

3) ACCESS REQUEST RATE (ARR) ALGORITHM

The ARR in Algorithm 2 takes two inputs—the access result and the UserID—and outputs the fuzzy value, ARR_{fuzzy}. The algorithm first checks whether the UserID exists in the ledger state and exits if it does not. Otherwise, the user is obtained from the chain ledger state and its attributes, StartSession, LastActiveTime, FirstReqTime, IdleTime, and CurrentTime, are obtained. The list is updated by adding a new access result using the loop count for permitted and denied requests. Following this, the time period is calculated, and the fuzzy function is called to update the ARR_{fuzzy}.

E. PROPOSED FUZZY INFERENCE SYSTEMS (FIS)

This section presents the design of the FIS for Reputation and ARR, the aim of which is to enhance the decision-making capabilities and trustworthiness of the blockchain-based access control system [4].

1) ACCESS REQUEST RATE FIS (ARR-FIS)

ARR represents the frequency or intensity at which users or entities attempt to access a specific resource, such as a file, database, or network. It can be calculated mathematically or using the FIS through the ARR mathematical equation presented in Equation (1). This does not accommodate all access requests. Thus, using the FIS to calculate the ARR handles the uncertainty and imprecision of data. Specifically, it captures and models the nonlinear relationships between the input and output variables for the ARR. Fig. 3 depicts the Membership Functions (MFs) linguistic variables for the ARR; the proposed system uses a Mamdani fuzzy method with a triangular membership function for all linguistic variables. The FIS uses three crisp inputs, permitted access requests (three MFs), denied access requests (three MFs), and the time period (three MFs), and gives the crisp output ARR (two MFs). To calculate the time period of committed access requests accurately, Table 2, presents the variables used in the time period equation, and Equation (2) displays the equation used for the time period. The calculation of the time period excludes the idle time when users do not request access; Equation (2), for different access requests. In the first case, when it was initially requested, the LAT and idle parameters were set to zero. This was followed by two requests, where the idle time was calculated using Equation (5) appeared. The old time is calculated using (3) and the active period is calculated using (4) to eliminate the idle period time by (5). An example of fuzzy logic rules is presented in Table 3. The overall ARR-FIS was designed using the MATLAB software.

$$\text{ARR} = \frac{\text{Total Access Requests}}{\text{Time Period}} \tag{1}$$

Algorithm 3 Proposed Access Control ABI

Inputs: Resource, Action, Reputation, Time.
Outputs: Results, Penalty, NoFR.
Require: *policycheck* ← *false*, *behaviorCheck* ← *true*, *RepCheck* ← *true*, *penalty* ← 0, *JC instancejudge*,

policy list policies, *TimeOfUnBlock of resource*
if this request is from the subject, **then**
  $p \leftarrow policies[resource][action]$
  $RepBool \leftarrow ReputationCheck(Reputation)$
  if Reputation is true **then**
    {Reputation Check}
    if p.policy = 'allow' **then**
      {static Check}
      $policyCheck \leftarrow true$
    else
      $policyCheck \leftarrow false$
    end if{Concurrent access Check}
    if $Time - p.ToLR \leq p.minInterval$ **then**
      $p.NoFR \leftarrow p.NoFR + 1$
      if $p.NoFR \leq p.threshold$ **then**
        Detect a misbehavior msb.
        $behaviorCheck \leftarrow false$.
        $penalty \leftarrow judge.misbehaviorJudge(subject, msb)$
        $TimeOfUnBlock \leftarrow time + penalty$.
        Push msb into the misbehavior list of resource.
      end if
    else
      $p.NoFR \leftarrow 0$
    end if
  end if
  if $TimeOfUnBlock \leq Time$ **then**
    {To initialize the NoFR}
    if $TimeOfUnBlock > 0$ **then**
      $p.NoFR \leftarrow 0, p.ToLR \leftarrow 0, TimeOfUnBlock \leftarrow 0$
    end if
  end if
  $p.ToLR \leftarrow time$
end if
$result \leftarrow policyCheckandRepCheck$.
$TriggereventreturnResult(result, NoFR, penalty, Reputation)$.

$$\text{Time Period} = \text{ActiveTime} + \text{OldTime} - \text{IdleTime} \tag{2}$$

$$\text{OldTime} = \text{LAT} + \text{FTR} \tag{3}$$

$$\text{ActiveTime} = \text{CT} - \text{SST} \tag{4}$$

$$\text{IdleTime} = (\text{SST} - \text{LAT}) + \text{OldIdle} \tag{5}$$

2) REPUTATION FIS (REPUTATION-FIS)

The main advantage of using fuzzy logic to infer reputation value is that it handles the subjectivity and uncertainty of reputation, which is influenced by various factors and subjective judgments. Fuzzy logic provides a framework to capture and model subjectivity and uncertainty by assigning membership values to reputation-related linguistic variables. This allows for the representation and processing of imprecise and subjective data, enabling a more nuanced assessment of reputation. Fig. 3b shows the reputation MFs. The proposed system uses the Mamdani fuzzy method with a triangular membership function for all linguistic variables. The FIS takes four crisp inputs: the old reputation value (five MFs), the ARR (two MFs), NoFR (two MFs), decay reputation (two MFs), and crisp output's new reputation (five MFs). Regarding the difference between the two crisp input reputation values, the old reputation results from the previous fuzzy logic evaluation, whereas the decay results after applying the RHDA algorithm to the old reputation. Hence, these results provide insight into how long the user has been inactive. NoFR is inferred from [4] and is used as a feedback variable for fuzzy logic. There are two variables related to NoFR: TimeOfUnBlock and penalty. Depending on the NoFR value, the penalty is calculated, based on the time of unblocking. NoFR was chosen as a crisp input for the FIS and the exclusion of the other two variables, TimeOfUnBlock and penalty, because NoFR has a specified range of [0,2], where zero means that no concurrent access has been detected, one means that the first concurrent access has been detected, and two means that concurrent accesses have been detected. It is difficult to determine the penalty and TimeOfUnBlock ranges, because the values increase exponentially and linearly. However, for the value updates of these variables, the penalty value increases only in cases where it is not set to zero. For TimeOfUnBlock and NoFR, the values are updated and set to zero when the user behaves normally. Hence, it was chosen as the NoFR crisp input due to its advantages, of specified range, value update, and initialization. A sample of the fuzzy logic rules is presented in Table 4. The overall ARR-FIS was designed using the MATLAB software.

TABLE 3. Samples of the arr fuzzy logic rule.

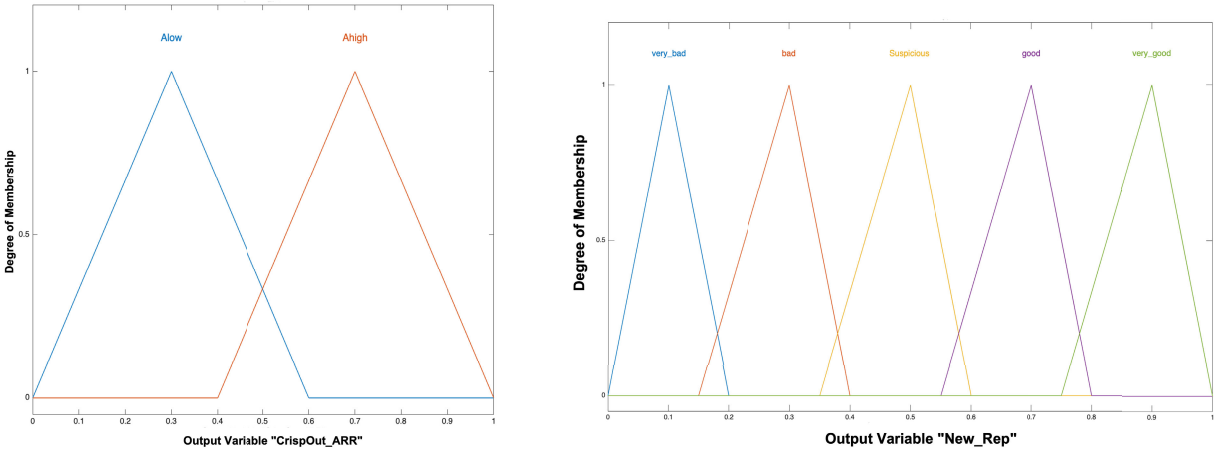
Rule	If Permitted	Denied	Time Period	Then ARR
1	Low	Low	Short	Low
2	Medium	Low	Short	Low
3	High	Low	Short	Low
4	Low	Medium	Short	High
5	Medium	Medium	Short	Low
6	High	Medium	Short	Low

TABLE 4. Samples of the reputation fuzzy logic rule.

Rule	If Rep_{Fuzzy}	ARR	NoFR	Then Rep_{Decay}	Then Rep_{New}
1	Very bad	Low	Bad	Bad	Very bad
2	Bad	Low	Bad	Bad	Very bad
3	Suspicious	Low	Bad	Bad	Very bad
4	Good	Low	Bad	Bad	Very bad
5	Very good	Low	Bad	Bad	Very bad
6	Very bad	High	Bad	Bad	Very bad

V. EVALUATION SETTINGS

This section describes the settings required before the framework evaluation. Furthermore, the reputation evaluation used in the proposed framework considers the access behaviors of users that are to be evaluated for each access request. The system parameters and factors used for the evaluation



(a) Membership function of ARR-FIS output.

(b) Membership function of Rep-FIS output.

FIGURE 3. Fig. (a) presents the Membership Functions (MFs) of ARR. It includes two MFs within the range [0,1], low, and high. Fig. (b) presents the MFs of reputation. It includes five MFs within the range [0,1], very bad, bad suspicious, good, and very good.

were inferred from [4] and [5]. To allow a comparative evaluation, they were set to the same values as those used in [4] and [5], as presented in Table 5. The parameters used for the proposed system are related to the user and are utilized to process the access request behavior using HyperLedger Fabric. The ledger state was then used to save updates of these parameters. To evaluate the prototype system, a firefly stack was created to deploy the HyperLedger Fabric blockchain with three nodes (three organizations) and interact with the Fabric through FireFly SDK. A Geth network was established with four nodes, two miner PCs, two light nodes, and a Raspberry Pi. For the Fabric blockchain, chaincode is deployed to evaluate the reputation and access request rate using fuzzy functions. Alongside this, several smart contracts are used to evaluate access requests in the Ethereum blockchain. However, the proposed framework uses different libraries, the two most important of which are Firefly SDK [32] and Web3 [33]. Although they are written in different languages—JavaScript and TypeScript—to exchange data between the two sides, the common module of JavaScript is used, where the TypeScript compiler initiates a JavaScript copy of the TypeScript file such that the firefly function is called inside the JavaScript. Additionally, to implement the FIS, a JavaScript library, FuzzyIS [34], was used to implement the fuzzification and defuzzification of the linguistic variables. The following sections present the evaluation analysis of the proposed system, including the analyzes of the reputation and ARR, a comparative evaluation, and analyzes of the system's performance and security.

TABLE 5. Initialization of the parameters.

Parameters	Initial Value
σ	0.01
n_1	2
n_2	4
W_{init}	1
n_{init}	1
W_{th}	30 min
W_f	18 min
minInterval	100 s
Threshold	2
JC Base	2
JC interval	3
UserID	0xD140CD292...
Decay_Rep	0.5
Fuzzy_Rep	0.5
Rep_list	Empty
LastActiveTime	0
policyResult	Empty
AccessRequestsRate	0
requestStartSession	Empty
FirstReqTime	Empty
IdleTime	Empty
NOFR	0
Penalty	0

VI. EVALUATION ANALYSIS

This section assesses and analyzes the results obtained from the evaluation of the proposed system. It strives to objectively evaluate the performance, effectiveness, and efficiency of a solution in achieving its intended goals and objectives. The evaluation results are analyzed and discussed in the following subsections. First, the reputation analysis subsection presents the analysis results of the proposed reputation system. Second, the Access Request Rate analysis subsection presents an analysis of fuzzy logic utilization to infer the Access Request Rate. Third, we present a comparative analysis of the proposed system [4], [5]. Finally, we present a security analysis of the proposed framework.

A. REPUTATION ANALYSIS

This subsection presents the analysis of the proposed reputation system. We evaluated the behaviors of two user types: a benign user (Fig. 4a presents a requester terminal

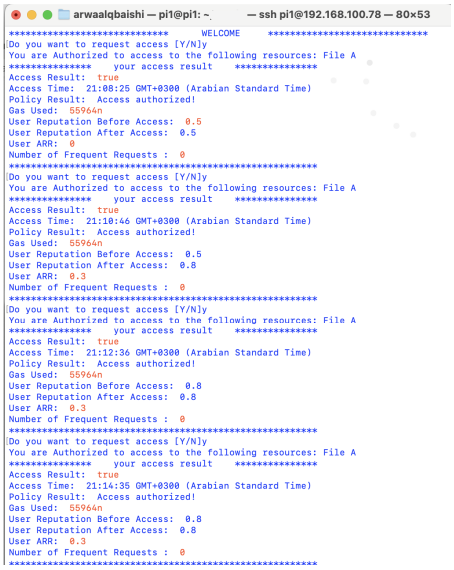
and Fig. 4b presents a monitor terminal) and a malicious user (Fig. 4c presents a requester terminal and Fig. 4d presents a monitor terminal). The evaluation includes the terminal output of the experiment from both the requester and monitor terminals. For benign users, access requests are authorized as long as they do not cause a concurrent request that may have a negative influence on their reputation value. Further, access requests originating from a malicious user are detected and prevented successfully. At 13:57, the user was in an inactive state with a low reputation, and their request for access was denied due to their bad reputation. Meanwhile, Rep-FIS updated the reputation value from 0.1 to 0.5. Hence, in the next access request at 13:57:28, the user was granted authorized access. However, this access request was detected as the first concurrent request. As a result, upcoming access requests from this user would be denied, and their reputation would be further degraded.

A comprehensive evaluation of both behaviors is shown in Fig. 5. In both cases, reputation starts at 0.5, the normal threshold, and then increases to 0.8. The evaluation shown in Fig. 5 depicts the reputation values after a several iterations. The evaluation began after the inactive states of the two scenarios. It can be seen that the benign user's reputation started at 0.8, whereas the malicious user's reputation started at 0.5. Furthermore, in the evaluation of a benign user (presented in Fig. 5), reputation remained steady, close to 0.8, based on their good history. The first arrow marks the beginning of the user's inactive state, during which time the RHDA updates the decay reputation variable. When the user becomes active, the fuzzy function processes two reputation values, a decay of zero and an old fuzzy reputation of 0.8, resulting in an output of 0.8. If the user mistakenly sends frequent access requests, the system accepts one instance of misbehavior. Thus, when a benign user causes the first concurrent access, the reputation value drops to a suspicious range as a countermeasure. However, it then recovers to the ideal range based on its good history after suspicious behavior. Conversely, the evaluation of malicious user (Fig. 5). We demonstrate that a user with a history of misbehaving access requests causes the reputation value to drop steeply to 0.5 in the suspicious area. The reputation value then continues to drop based on NoFR, in contrast to the decay reputation, which continues to drop based on the window value. Further, if there is a first instance of concurrent access, the value of the NoFR will be one, and the reputation will drop to the bad range of 0.3. If a second concurrent access request occurs, the value of the NoFR will be two, and the reputation will drop to a very bad range. The calculated penalty determines the recovery time, which is used to calculate the TimeOfUnBlock. For recovery, reputation will only increase to the suspicious range and will not increase based on past misbehavior.

Decay algorithms are designed to consider the recency and relevance of user actions, ensuring that the reputation values accurately reflect the most current and trustworthy information. As shown in the evaluation, window value was found to be positively related to reputation value. The time frame required for the RHDA to influence the reputation value depends on the window value, a larger window value implies that a longer time frame is needed to have an influence. During the experiment, for both use cases, as long as the user's reputation was greater than or equal to 0.5 (the reputation threshold), the initial window value would start at 1 and slowly increase. Once the window value would enter a ''fast increase'' stage at 18, it will continue increasing until it reached 30 (the window threshold). Then, the window value remained steady at 30, as long as the user's reputation remained at 0.5 or above. Whenever the user becomes inactive, the RHDA would negatively influence the reputation based on the window value. There are two stages of RHDA: the lower bound and the upper bound, where the window value has a multiplication relationship with the two stages. Further, the DEWA would also start to slowly decrease the window value until it reaches its threshold. If the period of inactivity falls within the range of the lower bound, the Rep_{Decay} value remains unchanged, indicating that the user may become active again within a short period. However, if the time period falls between the two stages, the Rep_{Decay} value starts to decrease linearly, as RHDA takes longer to negatively influence reputation. If the time period of inactivity exceeds the upper bound, Rep_{Decay} will decrease exponentially. On the other hand, if the user's reputation falls below the normal threshold, the window value rapidly drops to 1, causing RHDA to rapidly decrease the reputation value.

Moreover, Rep-FIS effectively evaluated the reputation value considering two use cases: benign and malicious users under different scenarios. Two scenarios were demonstrated in the evaluation. The first scenario is the penalization process. Rep-FIS successfully demonstrated this process by downgrading the reputation value due to suspicious and malicious activities. For the benign user, a first suspicious activity caused Rep-FIS to downgrade its reputation from 0.8 to 0.5. For malicious users, the first suspicious activity caused Rep-FIS to downgrade its reputation value from 0.5 to 0.3. If the user continued to misbehave, the reputation would decrease further to 0.1. The second scenario was the recovery process. Rep-FIS successfully demonstrated this process, whereas the reputation value recovered after inactivity, suspicious, and malicious activities by using only fuzzy logic rules. In the benign use case, Rep-FIS recovered the reputation after user inactivity from zero to 0.8, and from the suspicious activity from 0.5 to 0.8. For the malicious user, Rep-FIS recovered the reputation value after the inactive state from 0 to 0.5, and from a misbehavior from 0.1 to 0.5.

In terms of the root-mean-square error (RMSE), after applying system validation for the fuzzy logic system [34], using a dataset of 100 random values yielded a good RMSE value. However, RMSE quantifies the average discrepancy between the membership values predicted by the fuzzy logic system and the corresponding target values. Thus, the RMSE value reflects the accuracy of the fuzzy logic system's predictions, where a lower RMSE indicates that its



(a) Benign user case: requester terminal.

(b) Benign user case: monitor terminal.

(c) Malicious user case: requester terminal.

(d) Malicious user case: monitor terminal.

FIGURE 4. Terminal outputs during the experiment of two scenarios benign and malicious users. For benign user case: The two terminals Fig. 4a and Fig. 4b present the reputation increasing and decreasing during different time periods. The requester terminal runs from the subject side which runs from a raspberry pi. The ARR value 0.3 infers the user good history of access request behavior. The monitor terminal presents the access request information from the object side which also runs from a Raspberry Pi. Whereas malicious user case: Fig. 4c The first presents the requester terminal; Fig. 4d The second presents the monitor terminal. Both Terminals present the effect of access misbehavior on reputation value whereas the incoming requests are proactively denied based on the Reputation.

predicted membership values are closer to the target values on average. This suggests that the system performed well in producing accurate predictions. Conversely, a higher RMSE implies that the fuzzy logic system's predicted membership values deviate more from the target values on average. This indicates a lower level of accuracy in system's predictions. The resulting RMSE of 0.437 for a fuzzy system with three range inputs, [0,30], [0,1], [0,3], [0,1], and an output range of [0,1] indicated that, on average, the predictions of the fuzzy system were inaccurate by approximately 0.437 units in the context of the output range. This value seems reasonable and suggests that the fuzzy system performs relatively well in predicting the output within specified ranges. Further, given the narrowness of the output range ([0,1]), an RMSE of 0.437 indicated that the predictions were generally close to the actual outputs. This suggests that the fuzzy system

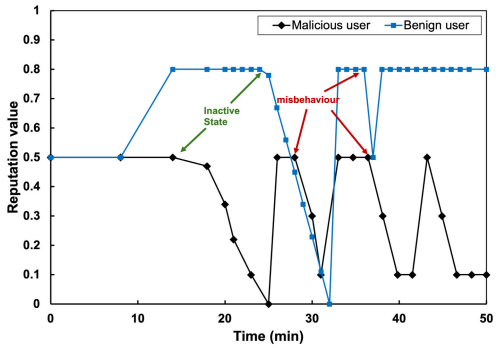


FIGURE 5. Malicious and benign chart. The chart depicts that RHDA Algorithm influenced the reputation values of both behaviors as shown in the green arrows. For misbehavior as shown in red arrows, a benign user reputation will remain in the range of very good even after one misbehavior where the framework accepts one mistake only when the user has a good access behavior history. Conversely, the malicious user reputation remains in the suspicious range. If they have a history of misbehavior. The reputation will continue to decline based on NoFR.

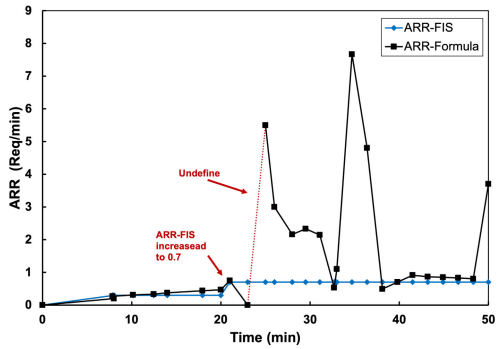


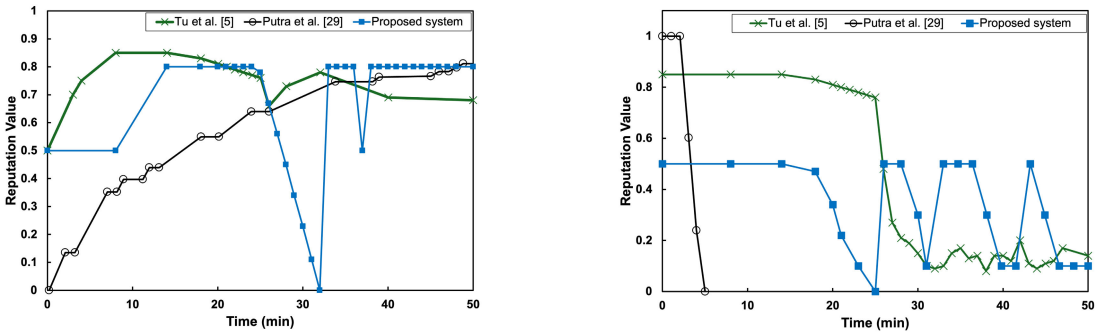
FIGURE 6. ARR analysis chart. The chart shows the gradual increase in the ARR formula. In contrast, the ARR-FIS remains stable within two specified ranges: 0.3 (Req/min) for the normal behavior access rate and 0.7 (Req/min) otherwise, as indicated by the First arrow. The second arrow highlights that the formula yields infinity regardless of the access request history within the time period. The ARR-FIS aims to address uncertainty and improve results by accounting for nonlinear relationships between input and output variables.

effectively captures the relationships between the inputs and outputs and that the predictions are relatively accurate within the specified ranges.

Overall, it is evident that benign user access behavior enhances reputation through consistent growth and progression, even after a period of inactivity. In contrast, malicious user access behavior reduces reputation and maintains the value in the suspicious range after a second malicious act. The evaluation of decay algorithms and their impact on user reputation has revealed a sophisticated and dynamic system that effectively captures the recency and relevance of user actions. The positive relationship between the window and reputation values, along with the intricate interplay between the RHDA and DEWA algorithms, demonstrates the system's ability to adapt to various user activity patterns. The findings of the experiment highlight the delicate balance between user inactivity and Rep_{Decay}, where the window value plays a crucial role in determining the time frame required for the RHDA to influence Rep_{Decay}. The system's ability to respond differently to user inactivity based on the lower and upper bounds, ensures that the Rep_{Decay} values accurately reflect the current trustworthiness of the user. Furthermore, the rapid reduction in the window value when a user's reputation falls below the normal threshold, triggering a swift decline in reputation, underscores the system's sensitivity to maintaining high standards of trustworthiness within the platform. Additionally, Rep-FIS has demonstrated its effectiveness in evaluating the reputation of benign and malicious users under different scenarios. The penalization process effectively downgraded the reputation of users based on suspicious and malicious activities, whereas the recovery process was able to restore reputation values after periods of inactivity or reduced misbehavior. By utilizing fuzzy logic rules, Rep-FIS was able to provide a nuanced and dynamic representation of user reputation, adjusting it accordingly based on the user's actions and access patterns. This comprehensive reputation management system shows promise in enhancing access control and security in IoT environments, where maintaining trust and accountability is crucial. The successful demonstration of these key scenarios highlights the potential of Rep-FIS as a valuable tool for managing and securing IoT environments. However, the RMSE for Rep-FIS of 0.437 with the specified input and output ranges is generally considered low, indicating that the fuzzy system performs well in predicting the output within the given ranges. In conclusion, the comprehensive analysis presented in this evaluation demonstrates the sophistication and effectiveness of the Rep-FIS system in preserving the integrity of user reputation, making it a valuable tool for fostering a reliable and trustworthy environment.

B. ACCESS REQUEST RATE (ARR) ANALYSIS

This subsection examines the methods used for Access Request Rates. However, as discussed in the methodology (in Section IV), the proposed system uses fuzzy logic instead of the mathematical equation. Fig. 6 presents the ARR chart. This includes a chart that displays a gradual increase in the ARR formula. Concomitantly, the ARR-FIS remained stable within two specified ranges, 0.3 Requests per minute (Req/min) for the normal behavior access rate and 0.7 (Req/min) otherwise, as indicated by the first arrow. Additionally, the second arrow indicates that the formula yields infinity, regardless of the access request history during the time period. Thus, the ARR-FIS resolves the problem of uncertainty and delivers significantly better results than the equation because it handles the nonlinear relationships between the input and output variables. The stable value of the ARR has a dynamic effect on the reputation value, and the combination of the ARR and NoFR is inversely associated with the reputation value. One drawback of the ARR formula is its undetermined range, which makes it difficult to quantify the range to be used as an input for the reputation fuzzy function.



(a) The evaluation under continuous normal behavior.

(b) The evaluation under continuous abnormal behavior.

FIGURE 7. Comparative evaluation chart of the proposed framework and other evaluation methods Tu et al. [5] and Putra et al. [29]. The chart presents both behaviors normal and abnormal, considering the inactivity status in the proposed framework and Tu et al. [5].

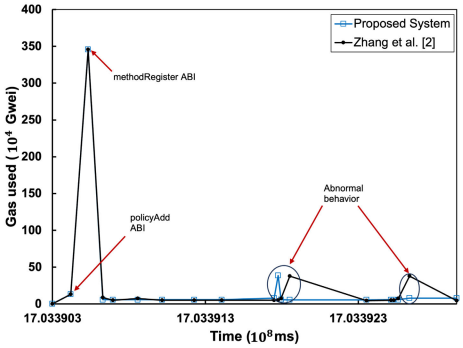


FIGURE 8. This chart compares the GasUsed of both proposed framework and [4], during smart contracts deployment and adding policies illustrated in first and second arrows. Moreover, the proposed framework saves more gas after the first abnormal behavior achieving the proactive approach, as shown in the third and fourth arrows.

C. COMPARATIVE EVALUATION

This subsection compares the proposed system with those presented in [4], [5], and [29]. The experiment simulated two users with different behaviors (presented in Fig. 7). For benign users (as shown in Fig. 7a, normal behavior.), the reputation values start at 0.5 for the proposed system and [5], while Putra et al. [29] the reputation value start at zero. Further, all trust models converged to a similar upper boundary, but with different convergence rates. Putra et al. [29] reached 0.8 at approximately 50 minutes. The proposed system and [5] converged to the upper bound of reputation value in less than 20 min, whereas the fuzzy-based reputation modeling in the proposed system contributed to this faster convergence for benign users, allowing them to quickly establish trust and gain access to the network. The reputation values remain unchanged as interactions continue consistently, demonstrating the convergence of reputation values for honest interactions. During inactive state, the reputation values for both the proposed system and [5], decreed gradually reflecting the recency of the reputation. Putra et al. [29] did not observe this feature. Putra et al. [29] found that reputation values were not influenced during an inactive state.

However, there is a limitation associated with the bootstrapping of new users with zero reputation values who may face challenges initially participating in the network initially [29]. Besides, both the proposed system and [5] have a faster convergence rate, which is generally less preferred, as it can make the model more vulnerable to newcomer attacks. For malicious interactions, the reputation values demonstrate a significant abrupt decline for malicious users, dropping to a lower bound within a specific number of time epochs. The reputation values dropped to zero for Putra et al. [29], and 0.1 for both the proposed system and [5], the reputation values dropped to zero. The results for all models show how reputation values rapidly decline for malicious users, indicating the effectiveness of the trust models in penalizing malicious behavior and protecting the network from malicious actors. The fuzzy-based reputation modeling in the proposed system may enable a more granular and responsive penalization of malicious behavior, quickly identifying and severely punishing such actions to protect the network.

The findings presented for the proposed system (as shown in Fig. 7a and Fig. 7b) confirm its consistency with the main features of the decay algorithm. The proposed system accurately evaluates long-term inactive users, such as those who misbehave after a long period of inactivity. The system classifies these users as suspicious based on their reputations, which drops them to the bad or very bad range following fuzzy processes. Even if the user behaves normally, their reputation remains in the suspicious range. Therefore, the proposed system was shown to be stable based on user input. In Fig. 8, presents the performance evaluation of the proposed system with that presented in [4]. A comparative evaluation is used to present the gas results used by specific transactions such as AccessControl ABI, MethodRegister ABI, and PolicyAdd ABI. The proposed system succeeded in providing a proactive approach by detecting the reputation value. If this is value is below the normal threshold, the access request is denied. Compared to the formal framework [4], where the user access request is checked based on TimeOfUnBlock, the

user is given access if the value is decreased or set to zero. As demonstrated in Fig. 8, the first and second arrows depict where the two systems spiked for deploying smart contracts and the policies were added; the third arrow shows the first instance of abnormal behavior where both systems spike, whereas in [4], only the last arrow spikes while the proposed system remains constant. Thus, the former delivers better results by proactively checking the reputation value before sending an access request. However, a similar pattern of results was obtained for both systems in most AccessControl ABI evaluations.

Overall, the proposed system achieved better outcomes, whereas the use of fuzzy logic solved the problem of uncertainty. The reputation value is assigned to the fuzzy set after the fuzzy logic conditions are met. In combination with the decay output, the proposed system provides a comprehensive evaluation of IoT users' reputations, which remains relatively constant over time within the specified range of the fuzzy set. This enables the network to maintain trusting relationships with users over a long period and improve its security against attacks. This suggests a tradeoff between quickly building trust for honest nodes and quickly detecting and penalizing malicious behaviors. The fuzzy-based approach enables more nuanced and adaptive handling of reputation decay, ensuring that the trust assessment remains accurate. Hence, the recency-based reputation decay feature of the proposed system [5] during inactive periods is an additional benefit compared to Putra et al. [29]. The analysis suggests that the proposed system and [5] have advantages in terms of faster convergence for benign users and recency-based reputation, while also effectively identifying and penalizing malicious behavior, although a faster convergence rate could be a potential drawback.

D. SECURITY ANALYSIS

This section analyzes the security of the proposed framework against various attacks. Furthermore, the framework adopts the same features as DEM-BTRM [4] and [5] because it employs two algorithms, RHDA and DEWA, for the access control model. Therefore, the proposed system can resist the following attacks:

- On–off attacks: Reducing the reputation evaluation interval and increasing the frequency of reputation evaluation prevents continuous malicious attacks.
- DoS attacks: The reputation value of malicious users who initiate attacks decreases rapidly when using a fuzzy logic system.
- Re-entry attacks: The proposed system stores the users' Geth IDs and reputation values on the blockchain by using RHDA. This avoids the problem of long-term inactive users, and the reputation converges to the initial reputation value, preventing users from attacking the network again after being inactive for a long time.
- Sybil attacks: The system stores evaluations of access request results in the blockchain along with user ID mapping, which makes it difficult for malicious users to initiate Sybil attacks to generate real and normal behaviors.

VII. DISCUSSION AND FUTURE WORK

The solutions presented in this article demonstrate promising results for evaluating user reputation. However, there are some potential limitations for future research. First, fuzzy logic can provide more nuanced and interpretable reputation assessments than crisp numerical values, and the complexity of the fuzzy system may make it more difficult for users to understand the reasoning behind the reputation values, resulting in a trade-off between the precise evaluation of IoT users' reputations and higher computational resources. The key limitations and computational overhead of the fuzzy algorithms used in the proposed system include the need to refine the fuzzy inference engine, particularly when aiming for comprehensive reputation evaluation, and the challenge of rapidly adapting to changing user behaviors and trust relationships within dynamic environments. Further, implementing and designing fuzzy systems can be complex, and the fuzzy rule base and membership functions may require frequent manual adjustments to maintain accurate trust evaluations. The complexity of the fuzzy inference engine scales with the number of fuzzy rules and membership functions employed, which can be computationally intensive, especially for large-scale systems with many users and interactions. Additionally, the use of decay algorithms to enhance accuracy and responsiveness by considering the recency and relevance of user actions may result in increased computational overhead due to the complexity of the fuzzy logic system and the decay algorithms.

Moreover, in a hybrid blockchain environment, the computational overhead of the fuzzy algorithms may impact the overall performance of the blockchain network, particularly in terms of transaction processing times, consensus mechanisms, and consume significant memory resources, which potentially limits the scalability of the system. To address these limitations, developing an adaptive learning mechanism to dynamically update the fuzzy rule base and membership functions, and combine fuzzy logic with other techniques. Several studies have driven further development by combining fuzzy logic with other techniques to overcome its limitations and improve the evaluation performance, such as Expert Judgment [14], game theory [13], and deep learning [19]. Second, the hybrid blockchain environment, in which the connection between two blockchain platforms poses potential security threats that must be mitigated. In particular, the exchanged data were not encrypted. To address this issue, investigating a cryptographic method to provide secure interactions within the IoT environment. Overcoming these limitations has the potential to further refine and expand the capabilities of the proposed system, ultimately leading to a more robust and efficient evaluation of user reputation. Finally, there are serious limitations associated with HyperLedger FireFly, particularly in terms of its high throughput. The `'MVCC_READ_CONFLICT'`

error indicates that multiple transactions are submitted concurrently to the Fabric network. These transactions may read or modify the same set of data or overlapping data items when multiple transactions access the same data simultaneously. Due to this potential limitation, the proposed system evaluates consecutive access requests individually, to temporarily avoid errors, and resolves the high throughput problem [31], providing the necessary flow capability to allow multiple parties to build sophisticated transaction flows collaboratively.

VIII. CONCLUSION

This article investigated the reputation evaluation of blockchain-based access control for the IoT. In this system, reputation is processed using fuzzy logic, and two algorithms are employed: DEWA and RHDA. Incorporating fuzzy logic handles the inherent vagueness and imprecision of reputation values and calculations of the access request rate. This leads to better and more consistent results compared to purely mathematical methods. On the other hand, the employment of decay algorithms ensures that values are up-to-date, thus providing a more reliable and robust reputation assessment. The proposed system was eventually deployed on two blockchain platforms—Hyperledger Fabric and Ethereum—which introduced a hybrid environment, leveraging the capabilities of both blockchain features with a more robust and versatile infrastructure, and overcoming the limitations of the original framework that relied solely on Ethereum.

A case study was presented to demonstrate the advantages of the proposed system. The proposed Rep-FIS has an RMSE of 0.437, indicating a high degree of accuracy. Furthermore, the proposed solutions maintain the integrity of the overall system, by dynamically detecting malicious attacks and proactively preventing future access requests from malicious users. For ARR-FIS, it was observed that the use of fuzzy logic in the ARR process yielded better results than the purely mathematical approach. The hybrid blockchain architecture provided lower gas costs than the previous Ethereum-only implementation, enhancing the overall efficiency of the system.

In conclusion, based on the empirical findings, the proposed system provides significant advancement by improving accuracy, enhancing security, and increasing cost-effectiveness compared to previous frameworks. However, there are some limitations of the current model including its reliance on preset fuzzy logic rules, which may not capture the full complexity of real-world scenarios. Additionally, the security of the hybrid blockchain environment requires further investigation to fully mitigate potential threats. In future work, the focus will be on improving the FIS process for both reputation and access request rates. Additionally, in the hybrid system environment, the connection of two blockchain platforms poses security threats that must be mitigated, by utilizing a cryptographic method to secure the transmission of data.

ACKNOWLEDGMENT

The authors extend their appreciation to the Deanship of Scientific Research at Imam Mohammad Ibn Saud Islamic University for funding this work through Graduate Students Research Support Program.

REFERENCES

[1] E. Schiller, A. Aidoo, J. Fuhrer, J. Stahl, M. Ziörjen, and B. Stiller, "Landscape of IoT security," *Comput. Sci. Rev.*, vol. 44, May 2022, Art. no. 100467. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1574013722000120

[2] S. Pal, A. Dorri, and R. Jurdak, "Blockchain for IoT access control: Recent trends and future research directions," *J. Netw. Comput. Appl.*, vol. 203, Jul. 2022, Art. no. 103371.

[3] A. I. A. Ahmed, S. H. Ab Hamid, A. Gani, S. Khan, and M. K. Khan, "Trust and reputation for Internet of Things: Fundamentals, taxonomy, and open research challenges," *J. Netw. Comput. Appl.*, vol. 145, Nov. 2019, Art. no. 102409.

[4] Y. Zhang, S. Kasahara, Y. Shen, X. Jiang, and J. Wan, "Smart contract-based access control for the Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1594–1605, Apr. 2019.

[5] Z. Tu, H. Zhou, K. Li, H. Song, and Y. Yang, "A blockchain-based trust and reputation model with dynamic evaluation mechanism for IoT," *Comput. Netw.*, vol. 218, Dec. 2022, Art. no. 109404.

[6] A. Kesarwani and P. M. Khilar, "Development of trust based access control models using fuzzy logic in cloud computing," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 34, no. 5, pp. 1958–1967, May 2022.

[7] H. Liu, D. Han, and D. Li, "Fabric-IoT: A blockchain-based access control system in IoT," *IEEE Access*, vol. 8, pp. 18207–18218, 2020.

[8] B. Chai, B. Yan, A. Dong, and J. Yu, "SFAC: A smart contract-based fine-grained access control for Internet of Things," *Proc. Comput. Sci.*, vol. 187, pp. 335–340, Jan. 2021.

[9] W. Xiang and Z. Yuanyuan, "Scalable access control scheme of Internet of Things based on blockchain," *Proc. Comput. Sci.*, vol. 198, pp. 448–453, Jan. 2022.

[10] C. A. Reyes-García and A. A. Torres-Garcia, "Fuzzy logic and fuzzy systems," in *Biosignal Processing and Classification Using Computational Learning and Intelligence*. Amsterdam, The Netherlands: Elsevier, 2022, pp. 153–176.

[11] Y. Liu, J. Wang, Z. Yan, Z. Wan, and R. Jäntti, "A survey on blockchain-based trust management for Internet of Things," *IEEE Internet Things J.*, vol. 10, no. 7, pp. 5898–5922, Apr. 2023.

[12] Z. Zulkifl, F. Khan, S. Tahir, M. Afzal, W. Iqbal, A. Rehman, S. Saeed, and A. M. Almuhaideb, "FBASHI: Fuzzy and blockchain-based adaptive security for healthcare IoTs," *IEEE Access*, vol. 10, pp. 15644–15656, 2022.

[13] C. Esposito, O. Tamburis, X. Su, and C. Choi, "Robust decentralised trust management for the Internet of Things by using game theory," *Inf. Process. Manage.*, vol. 57, no. 6, Nov. 2020, Art. no. 102308.

[14] H. F. Atlam, R. J. Walters, G. B. Wills, and J. Daniel, "Fuzzy logic with expert judgment to implement an adaptive risk-based access control model for IoT," *Mobile Netw. Appl.*, vol. 26, no. 6, pp. 2545–2557, Dec. 2021.

[15] B. B. Gardas, A. Heidari, N. J. Navimipour, and M. Unal, "A fuzzy-based method for objects selection in blockchain-enabled edge-IoT platforms using a hybrid multi-criteria decision-making model," *Appl. Sci.*, vol. 12, no. 17, p. 8906, Sep. 2022.

[16] P. Wang, N. Xu, H. Zhang, W. Sun, and A. Benslimane, "Dynamic access control and trust management for blockchain-empowered IoT," *IEEE Internet Things J.*, vol. 9, no. 15, pp. 12997–13009, Aug. 2022.

[17] X. Yuan, M. J. Liebelt, P. Shi, and B. J. Phillips, "Cognitive decisions based on a rule-based fuzzy system," *Inf. Sci.*, vol. 600, pp. 323–341, Jul. 2022.

[18] Z. Lian, P. Shi, C.-C. Lim, and X. Yuan, "Fuzzy-model-based lateral control for networked autonomous vehicle systems under hybrid cyber-attacks," *IEEE Trans. Cybern.*, vol. 53, no. 4, pp. 2600–2609, Apr. 2023.

[19] A. Yazdinejad, A. Dehghantanha, R. M. Parizi, G. Srivastava, and H. Karimipour, "Secure intelligent fuzzy blockchain framework: Effective threat detection in IoT networks," *Comput. Ind.*, vol. 144, Jan. 2023, Art. no. 103801.

[20] A. H. Lone and R. N. Mir, "Reputation driven dynamic access control framework for IoT atop poa Ethereum blockchain," *IACR Cryptol. ePrint Arch.*, vol. 2020, p. 566, Jan. 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:218654526

[21] D. Wu and N. Ansari, "A trust-evaluation-enhanced blockchain-secured industrial IoT system," *IEEE Internet Things J.*, vol. 8, no. 7, pp. 5510–5517, Apr. 2021.
[22] H. Song, Z. Tu, and Y. Qin, "Blockchain-based access control and behavior regulation system for IoT," *Sensors*, vol. 22, no. 21, p. 8339, Oct. 2022.
[23] M. Li, H. Zhou, and Y. Qin, "Two-stage intelligent model for detecting malicious DDoS behavior," *Sensors*, vol. 22, no. 7, p. 2532, Mar. 2022.
[24] W. Jiang, Z. Lin, and J. Tao, "An access control scheme for distributed Internet of Things based on adaptive trust evaluation and blockchain," *High-Confidence Comput.*, vol. 3, no. 1, Mar. 2023, Art. no. 100104.
[25] Y. Yang, Z. Tu, Y. Liu, and H. Zhou, "Blockchain-empowered token-based access control system with user reputation evaluation," *Comput., Mater. Continua*, vol. 77, no. 3, pp. 3163–3184, 2023.
[26] A. A. Arsyad, I. W. Widayat, and M. Köppen, "Supporting farming smart documentation system by modular blockchain solutions," *Decis. Making, Appl. Manage. Eng.*, vol. 5, no. 1, pp. 1–26, Mar. 2022.
[27] M. Sun, "The application of embedded hardware system and blockchain in rural financial management cloud platform," *Decis. Making: Appl. Manage. Eng.*, vol. 7, no. 2, pp. 81–100, Feb. 2024.
[28] S. Malik, V. Dedeoglu, S. S. Kanhere, and R. Jurdak, "TrustChain: Trust management in blockchain and IoT supported supply chains," in *Proc. IEEE Int. Conf. Blockchain (Blockchain)*, Jul. 2019, pp. 184–193.
[29] G. D. Putra, V. Dedeoglu, S. S. Kanhere, R. Jurdak, and A. Ignjatovic, "Trust-based blockchain authorization for IoT," *IEEE Trans. Netw. Service Manage.*, vol. 18, no. 2, pp. 1646–1658, Jun. 2021.
[30] G. D. Putra, V. Dedeoglu, S. S. Kanhere, and R. Jurdak, "Trust management in decentralized IoT access control system," in *Proc. IEEE Int. Conf. Blockchain Cryptocurrency (ICBC)*, May 2020, pp. 1–9.
[31] *Fabric—Hyperledger.Github.Io*. Accessed: Jan. 2, 2024. [Online]. Available: https://hyperledger.github.io/firefly/tutorials/custom_contracts/fabric.html
[32] Firefly. *GitHub Hyperledger/Firefly-Sdk-Nodejs: FireFly SDK for Node.js Github.com*. Accessed: Jan. 2, 2024. [Online]. Available: https://github.com/hyperledger/firefly-sdk-nodejs
[33] S. M. Jain, "Hardhat," in *A Brief Introduction to Web3: Decentralized Web Fundamentals for App Development*. Berkeley, CA, USA: Apress, 2023, pp. 167–179, doi: 10.1007/978-1-4842-8975-4_8.
[34] M. Nesterenko. *GitHub Maksnester/FuzzyIS: Lib Describes Core Fuzzy Logic Things for Building Fuzzy Inference Systems. Github.Com*. Accessed: Jan. 2, 2024. [Online]. Available: https://github.com/maksnester/fuzzyIS

ARWA A. ALQBAISHI received the B.S. degree in computer science and the M.S. degree in information security from the University of Imam Mohammad Ibn Saud Islamic, Riyadh, Saudi Arabia. Her research interests include artificial intelligence and cryptography, access control, blockchain technology, and network security.

ALAA E. S. AHMED received the Ph.D. degree in computer science and engineering from the University of Connecticut, USA, in 2005. His Ph.D. thesis was on fault tolerant real time scheduling. He works as an Assistant Professor with the College of Computer Science and information, Imam Mohammad Ibn Saud Islamic University. His research interests include information security, cloud computing environment, and wireless sensor network performance evaluation.