IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 11 October 2023, accepted 21 November 2023, date of publication 30 November 2023,
date of current version 8 December 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3337806

RESEARCH ARTICLE

Blockchain-Based Trust Management for Virtual Entities in the Metaverse: A Model for Avatar and Virtual Organization Interactions

KAMRAN AHMAD AWAN[1], IKRAM UD DIN[1], (Senior Member, IEEE), AHMAD ALMOGREN[2], (Senior Member, IEEE), AND BYUNG SEO-KIM[3], (Senior Member, IEEE)

[1]Department of Information Technology, The University of Haripur, Haripur 22620, Pakistan
[2]Department of Computer Science, College of Computer and Information Sciences, King Saud University, Riyadh 11633, Saudi Arabia
[3]Department of Software and Communications Engineering, Hongik University, Sejong-si 30016, South Korea

Corresponding author: Byung Seo-Kim (jsnbs@hongik.ac.kr)

This work was supported in part by the National Research Foundation (NRF), South Korea, through the Project BK21 FOUR; and in part by King Saud University, Riyadh, Saudi Arabia, through the Researchers Supporting Project RSP2023R184.

ABSTRACT As blockchain technology and decentralized systems evolve, the security of these infrastructures faces challenges from increasingly sophisticated threats. This research introduces a methodology designed to strengthen the security parameters of distributed systems, with a specific focus on its applicability within the Metaverse. Our probabilistic trust model dynamically allocates weights to system nodes based on their observed behaviour and the reputation of associated entities. This mechanism effectively counters a range of security threats, including the Sybil, Good/Bad mouthing, and On/Off attacks. By integrating blockchain technology, we establish a robust trust foundation within the Metaverse, ensuring enhanced security for digital interactions. To further combat deceptive activities and reduce superfluous intermediaries, our model incorporates smart contracts. Beyond their transactional utility, these contracts function as trust regulators for interactions among Metaverse avatars. Our trust model efficiently differentiates various virtual entities, assigning trust scores that resonate with their specific classifications. We also introduce a decentralized dispute resolution framework, where virtual entities act as impartial arbiters, promoting transparency and fairness in conflict resolution. We have implemented our proposed solution on real-time blockchain platform in comparison with the existing appraoches, i.e., BTCGS and MSBC-CTrust. The evident enhancements in threat detection capabilities and the agility in neutralizing these threats validate our model's resilience and adaptability.

INDEX TERMS Metaverse, virtual environment, blockchain, trust management, virtual environment, security, privacy preservation, reputation management, avatar trust.

I. INTRODUCTION

The rapid progression of the metaverse [1] brings to the fore the critical concern of establishing and preserving trust amongst its virtual entities, including avatars and virtual organizations [2], [3]. Traditional trust management strategies, dependent on central bodies or third-party intermediaries, are ill-suited to the decentralized nature of the metaverse [4], highlighting the exigent need for a robust, decentralized trust system resistant to cyber threats and ensuring transparency and accountability for all participants. Blockchain technology, with its decentralized foundation and transparent operational ethos, stands out as a viable solution to this challenge [5]. However, employing blockchain for metaversal trust management remains a nascent research area. The transformative properties of blockchain and its decentralized

The associate editor coordinating the review of this manuscript and approving it for publication was Deepak Mishra.

networks, encompassing transparency, immutability, and the elimination of intermediaries, have fundamentally reshaped transactional and informational exchange paradigms. The ethos of decentralization has spurred the development of distributed applications and platforms, transferring control from singular entities to node networks. Despite their advantages, decentralized systems confront security challenges, with malicious entities continually attempting to exploit vulnerabilities.

Recognizing the profound capabilities of distributed systems and blockchain, this paper aims to present measures to strengthen their security profiles. Our primary focus lies in exploring the potential of blockchain for trust management within the metaverse and devising a trust management framework for virtual entities. Within the metaverse, trust is integral to facilitating user interactions and transactions with other virtual entities. A deficiency in trust can hinder user engagement and limit potential opportunities, particularly in an environment devoid of physical interactions. As the metaverse continues to mature, the need for decentralized trust management for its virtual entities becomes increasingly evident [6]. Conventional trust mechanisms, predicated on intermediaries or central bodies, fall short in the metaversal context due to their inherent centralization, heightened costs, and attack vulnerabilities. Consequently, a transparent, decentralized trust system is paramount for secure exchanges between digital entities in the metaverse. The foundation of social and economic metaversal interactions lies in trust between avatars and virtual organizations. Implementing distributed trust systems [7] for the metaverse demands a transparent, decentralized approach, a pursuit fraught with complexities. Blockchain, by maintaining immutable and transparent records, presents an avenue to foster trust in virtual spaces without the crutch of intermediaries or central authorities [8].

With the advent of the metaverse, the necessity for a robust trust management system becomes paramount. This paper proposed a model that capitalizes on the intrinsic qualities of blockchain technology, such as transparency and immutability, to provide a trusted, secure, and decentralized trust management system for virtual entities. Our model integrates smart contracts, reputation systems, and a decentralized dispute resolution mechanism to foster a secure environment for these entities. Trust ratings, assigned to virtual entities, are adaptive, reflecting the ever-evolving metaverse based on individual attributes and actions. Through the utilization of smart contracts, encoded directly on the blockchain, trust agreements between entities are automated, ensuring authenticity, transparency, and mitigating intermediary intervention and potential fraud. The reputation system, by providing accessible ratings of each entity, promotes virtuous behavior, thereby enriching the metaversal community. Additionally, the model encompasses a decentralized dispute resolution mechanism, ensuring equitable and transparent conflict resolution. Our contributions can be outlined as:

1) Utilization of blockchain technology tailored for trust management in the metaverse, crafting a reliable environment for virtual entities.
2) Adaptation of smart contracts to facilitate trust agreements, thereby eradicating intermediaries. Despite the widespread application of smart contracts in various blockchain systems, our model uniquely applies them to manage trust amongst virtual entities in the metaverse.
3) Recognition of the diverse nature of metaverse entities, allowing the system to discern between different entity types and apportion trust levels accordingly.
4) Introduction of a decentralized dispute resolution feature, appointing select entities as arbitrators, ensuring a transparent, decentralized conflict resolution process.

The remainder of this article is structured as follows: Section II delineates related research. Section IV elucidates our methodology for countering specific attacks on reputation-centric trust systems, such as the Good Mouthing Attack, Bad Mouthing Attack, and Sybil Attack, using tools like reputation-driven voting and trust metrics. Experimental simulations are detailed in Section V. The concluding remarks are presented in Section VIII.

II. RELATED WORK

The Metaverse represents a novel digital environment, allowing users to immerse themselves in a myriad of activities spanning gaming, social interactions, education, and commerce. Its promise is undeniable, aiming to redefine our engagement with digital platforms. Nevertheless, the establishment of trust in the Metaverse is pivotal and has garnered significant academic attention. A comprehensive comparison of various research studies that delve into the intersections of Blockchain and Metaverse technologies is essential for understanding their contributions, limitations, and potential future directions. Table 1 provides a detailed comparative analysis of notable contributions in this domain, elucidating their primary focus, inherent challenges, proposed future research trajectories, and key remarks.

Zhang et al. [9] delved into trust-building mechanisms, assessing their influence on purchase intentions within Metaverse shopping scenarios. Their findings posit a direct positive correlation between trust and purchase intent. Intriguingly, age emerged as a moderator, underscoring the need for age-sensitive trust-building strategies. On a parallel tangent, Ali et al. [10] advocated for the amalgamation of Explainable AI and Blockchain within the Metaverse, specifically for bolstering trust and ensuring data security in healthcare applications. Their proposition centers on facilitating virtual interactions between patients and healthcare providers, emphasizing the critical role of trust in healthcare transformations through the Metaverse.

Sathya [11] accentuated the role of Blockchain technology, presenting it as a cornerstone for Metaverse trust. The research accentuates the indispensability of decentralized

TABLE 1. Comparative analysis of blockchain and metaverse approaches.

Ref.	Contribution	Limitation	Future Direction	Remarks
[9]	Proposed a trust mechanism to bolster purchase intent in Metaverse retail scenarios	Specific to Metaverse retail; lacks wider applicability	Investigate its applicability to diverse Metaverse sectors	A notable exploration into trust in Metaverse retail contexts
[10]	Merged Explainable AI and Blockchain for enhanced healthcare data protection within the Metaverse	Primarily tailored for healthcare; lacks broad applicability	Extend to diverse Metaverse domains	A pivotal work on healthcare data security within the Metaverse
[11]	Elucidated the intrinsic role of Blockchain in fostering Metaverse trust	Primarily theoretical; lacks empirical validation	Emphasize empirical evaluations in subsequent studies	A theoretical examination of Blockchain's role in the Metaverse
[12]	Introduced a trusted Blockchain-driven governance system for Metaverse manufacturing	Exclusively targets manufacturing; demands broader validation	Probe its viability in diverse Metaverse industries	A seminal piece on applying Blockchain to Metaverse manufacturing
[13]	Advocated for a unified Metaverse framework utilizing Trusted AI and Blockchain	Experimental scope is narrow; demands broader empirical investigation	Encourage expansive real-world evaluations	A study delineating Trusted AI and Blockchain synergy
[14]	Devised a Blockchain-anchored multisignature lock mechanism for Metaverse UAC	Requires exhaustive security validation	Emphasize rigorous security evaluations in subsequent endeavors	Pioneering work on Metaverse user access control
[15]	Delivered a comprehensive review on Blockchain-driven trust management strategies for IoT	Presents an overview; doesn't introduce novel methodologies	Inspire novel system proposals based on the survey insights	A comprehensive review of Blockchain in IoT trust management
[16]	Crafted a Master-Slave Blockchain architecture for holistic cross-domain trust management	The system's scalability remains uncertain	Address scalability concerns in future iterations	A groundbreaking blueprint for cross-domain trust management
[17]	Presented a secure data-sharing paradigm for IoT, focusing on agricultural risk management using Blockchain	Solely targets agriculture; requires broader validation	Widen its applicability to diverse IoT sectors	An innovative application of Blockchain in agricultural IoT

trust paradigms, with Blockchain heralded as an instrument for fostering such trust. Similarly, Lin et al. [12] introduced a Blockchain-based trustworthy governance framework tailored for Metaverse production scenarios. Their methodology seeks to enhance trustworthiness in Metaverse manufacturing engagements, underscoring the centrality of trust in this domain.

Adding to this narrative, Badruddoja et al. [13] championed the integration of Blockchain and Trusted AI, aiming to amplify the Metaverse's capabilities. Their blueprint is anchored in fostering secure and credible interactions with digital assets. Gai et al. [14] put forth a multi-signer lock mechanism for user access controls in the Metaverse, built upon Blockchain foundations. Their focus remains steadfast on trust as a mechanism for bolstering Metaverse security and privacy.

Broadening the scope to the Internet of Things (IoT), Liu et al. [15] conducted an extensive survey on Blockchain-mediated trust management. Their work underlines the prospective synergy between Blockchain and the enhancement of trust within IoT ecosystems. Wu et al. [16] proposed a unique Primary-Secondary Blockchain architecture equipped with a cross-domain trust ticket. Their architectural design aspires to manage trust across disparate domains within the Metaverse, emphasizing trust's role in safeguarding cross-domain transactions. Lastly, Manoj et al. [17] proposed a secure framework for IoT data sharing, integrating oracle-based access controls tailored for agricultural risk management. Their solution endeavors to preserve the integrity and confidentiality of agricultural data within the Metaverse, highlighting trust as an instrumental factor in Metaverse-driven agricultural endeavors.

III. BLOCKCHAIN AND TRUST MANAGEMENT IN THE CONTEXT OF THE METAVERSE

The Metaverse, an expansive and decentralized virtual shared space, has rapidly emerged as a nexus of technological evolution, encapsulating advancements in virtual reality, blockchain, and decentralized systems. Ensuring trust within this boundless digital realm becomes pivotal. While trust management in decentralized systems has been extensively studied, its intricacies within the Metaverse remain relatively uncharted. Moreover, the synergy between blockchain and trust management presents a promising solution. This section delves into a comprehensive review of recent works that touch

TABLE 2. Comparison of referenced articles concerning trust and security mechanisms in the context of the Metaverse.

Ref.	Study Domain	Trust & Security Focus	Suitability/Challenges for Metaverse
[10]	Healthcare	Data security	Integration with Metaverse platforms
[18]	Metaverse	Trustless architecture	Direct applicability
[19]	Metaverse & Multitasking	security	Leveraging federated learning
[20]	Tourism	Blockchain adoption	Transfer of trust mechanisms
[21]	Industry	Blockchain integration	Scaling in Metaverse
[22]	Metaverse	Digital asset trust	Narrow trust domain
[23]	Metaverse	Consensus mechanisms	Scalability issues
[24]	Metaverse	Cross-chain interaction	Potential synchronization issues
[25]	IoT	Cold-start trust management	Broader trust fabric

upon these themes, seeking to understand their relevance and potential limitations in the context of the Metaverse.

Ali et al. [10] present an intricate fusion of the Metaverse with healthcare, emphasizing the importance of trust. Their work brings forth the healthcare domain as a testament to the potential applications of the Metaverse beyond gaming and social interaction. As healthcare data is critically sensitive, the integration of explainable AI and blockchain is proposed to ensure data security and enhance trust. The research underscores a pivotal aspect: trust is not just about transactions or interactions but also about understanding and explaining processes to users, especially in domains where stakes are high, such as medical decision-making. However, while the paper beautifully stitches the narrative of Metaverse in healthcare, it focuses more on a theoretical synthesis, with less emphasis on the practical challenges that might arise in deploying blockchain-based trust mechanisms in a healthcare-centric Metaverse.

Xu et al. [18] venture into the realm of designing a blockchain-enabled Metaverse that operates on trustless principles. The term 'trustless' in blockchain parlance refers to the notion that interactions occur without participants needing to trust each other, thanks to the immutable and transparent nature of blockchain transactions. The authors meticulously unravel the layers of how trustless operations can be seamlessly integrated into a Metaverse. However, one might argue that the Metaverse, by its inherent design and purpose, is more than just transactions. The emotional, social, and experiential aspects of user interactions might demand a more nuanced approach to trust than what a purely trustless architecture can offer. The approach of Moudoud and Cherkaoui [19] stands as an emblem of the potential amalgamation of advanced learning techniques with blockchain for the Metaverse. By integrating multi-tasking federated learning with blockchain, they envision a Metaverse where trust and security are fostered organically. Their approach hints at the adaptability of decentralized systems, where learning from data can potentially enhance the trustworthiness of interactions. However, the Metaverse's dynamic nature might pose challenges in ensuring that federated learning models are always up-to-date and reflective of the continuously evolving trust dynamics.

Corne et al. [20] offer an intriguing perspective on the marriage of blockchain technology with the tourism sector and further its implications for the Metaverse. The paper underlines the crucial determinants for adopting blockchain in tourism and extrapolates this to envision Metaversal applications. By bridging the physicality of tourism with the digital expanse of the Metaverse, this study offers a unique lens. However, one might question the direct applicability of determinants from a sector as tangible as tourism to the fluid, dynamic Metaverse, which operates on a spectrum of different interactional paradigms. Mourtzis et al. [21] delve deep into the possibilities of blockchain technology in an industrial setting within the Metaverse. As industries pivot towards this new-age digital ecosystem, the importance of secure, transparent, and tamper-proof systems increases manifold. Their paper highlights the paradigm shifts in industrial processes due to the blockchain's introduction. Yet, the challenges of scaling, interoperability, and real-time synchronization in an industrial Metaverse context need further exploration.

The approach by Islam and Tan [22] zeroes in on the burgeoning domain of digital assets within a Web 3 based Metaverse. The inherent value and transferability of digital assets necessitate a trust mechanism. Their insights into how blockchain can underpin these trust requirements are enlightening. However, while the paper accentuates the transactional aspect of trust, it might benefit from addressing the experiential facets of trust interactions within a diverse Metaversal community. Rajawat et al. [23] tackle the twin challenges of security and scalability in the Metaverse using blockchain-based consensus mechanisms. By ensuring that all nodes in the network agree upon the truthfulness of transactions, the paper posits an enhanced security posture. The dynamic and expansive nature of the Metaverse might necessitate an adaptive consensus approach that the paper could further explore.

Ren *et al.'s* [24] work on HCNCT provides an architectural foundation for a blockchain-based Metaverse. By introducing a cross-chain interaction scheme, the paper addresses some critical challenges regarding interoperability between different blockchain networks. This endeavor paves the way for a unified, cohesive Metaverse. However, practical challenges

in terms of transaction latency and chain synchronization might emerge as potential bottlenecks. TMETA, as proposed by Wang et al. [25], champions trust management for IoT services using a digital twin aided blockchain mechanism. The paper's approach of simulating physical IoT entities in the digital space and managing trust dynamics can be revolutionary. But the Metaverse, with its vast array of interactions, might demand an even more intricate trust fabric, expanding beyond the IoT-centric viewpoint.

IV. PROPOSED METHODOLOGY

This section introduces the proposed methodology for a blockchain-anchored trust management system tailored for the metaverse. The emphasis is on creating a steadfast platform enabling secure interactions for virtual entities. The methodology integrates smart contracts to streamline trust agreements and eliminates intermediaries. A diversified trust model is introduced, accounting for the varied nature of virtual entities, supplemented by a decentralized dispute resolution system. A salient feature of our approach is the management of Avatar Trust and Reputation. In the metaverse, avatars function as the primary interface, influencing interaction quality based on their trustworthiness. The proposed model assigns trust ratings to avatars based on historical behaviour and peer evaluations. This trust is quantified through Reputation Management, which periodically reviews an avatar's actions. Positive behaviours bolster reputation, while negative actions detrimentally affect it. The combined effects of Avatar Trust and Reputation Management amplify the security and trustworthiness within the metaverse.

In decentralized systems, particularly those intrinsic to the Metaverse, entities and their interactions inherently exhibit characteristics of variability and uncertainty. This unpredictable nature of behaviors calls for a trust model that does not merely rely on deterministic, fixed parameters. Instead, a more nuanced approach is needed, one that can adapt to the fluidity of such environments. The proposed dynamic, probabilistic network-oriented trust model has been formulated with these considerations at its core. Traditional deterministic models for trust evaluation, which are grounded on fixed variables and static conditions, often struggle to accurately represent or predict behaviors in the Metaverse. Such models might be effective in environments where behaviors are consistent and predictable, but they become less reliable when applied to the unpredictable Metaverse ecosystem. The core of the proposed trust model embraces a probabilistic approach, allowing for an adaptive representation of trust that takes into account the uncertainties associated with each node's behavior. By doing so, the model becomes more resilient and adaptive, ensuring that trust evaluations are timely and reflective of the current state of the network, rather than being rooted in historical or static data. Furthermore, by being network-oriented, the model recognizes the interdependencies and relational dynamics between nodes, thereby providing a more comprehensive and holistic view of trust in the system.

Historically, trust models in decentralized systems relied on fixed weight allocation mechanisms, predominantly based on past interactions and static attributes. Such approaches, while effective in relatively consistent environments, exhibit limitations when confronted with the dynamic nature and evolving threats present within the Metaverse. These traditional mechanisms often struggle to promptly adapt to abrupt behavioral changes or recognize the significance of external affiliations in trust evaluation. The inherent dynamism and complexity of the Metaverse demand a more adaptable trust assessment mechanism. Our model's distinct approach to weight allocation is rooted in two primary considerations: real-time behavior of nodes and the reputation of affiliated entities.

- **Behavior-based Weight Allocation:** By dynamically adjusting weights according to the recent behaviors of nodes, our model maintains a timely and accurate representation of each node's trustworthiness. This ensures that sudden deviations from expected behavior patterns, even from historically reliable nodes, do not go unnoticed or unaddressed.
- **Incorporating Affiliated Entity Reputation:** An entity's reputation within its affiliated group or organization can provide crucial context to its behaviors within the broader network. By integrating this affiliated reputation into the weight allocation process, the model can achieve a more nuanced and comprehensive trust assessment, effectively bridging the gap between individual node behaviors and larger organizational dynamics.

To elucidate the advantage of our approach, consider the following scenario: Suppose a node, which has consistently demonstrated reliability in past interactions, suddenly exhibits anomalous behaviors. Traditional weight allocation mechanisms, grounded in historical data, might fail to detect or adequately respond to this behavioral shift. In contrast, our proposed model, with its dynamic weight adjustments factoring in both current behavior and the reputation of the affiliated entity, is primed to detect such anomalies and take appropriate, timely measures to address potential threats.

A. ARCHITECTURAL FRAMEWORK OF PROPOSED METHODOLOGY

The methodology harnesses blockchain to architect a trust-centric environment for the metaverse's virtual entities. Key components of the architectural framework are depicted in Figure 1.

1) **Blockchain-based Trust Management:** The system deploys a blockchain network to underpin the trust management framework. Trust-related records, encompassing ratings, agreements, and dispute resolutions, are preserved indelibly on the blockchain, fortifying transaction transparency.
2) **Automated Trust Agreements and Intermediary Removal:** Trust agreements in the metaverse are automated through smart contracts, eliminating

intermediary intervention, reducing fraud potential and bolstering system efficiency.

3) **Heterogeneous Trust Model:** The proposed model mirrors the multifaceted nature of metaverse's virtual entities. It discriminates between entities based on inherent attributes and behaviours, allocating trust ratings appropriately.
4) **Decentralized Dispute Resolution Mechanism:** The system integrates a decentralized approach to dispute redress. A designated cohort of virtual entities functions as arbitrators, ensuring resolutions are fair and devoid of centralized bias.

B. BLOCKCHAIN-ORIENTED TRUST MANAGEMENT

The metaverse's flourishing ecosystem relies heavily on the ability to foster trustworthy interactions. Trust, in this context, serves as the bedrock of the engagements between the metaverse's virtual entities. In this study, we propose a methodology that leverages the functionalities of blockchain technology to architect a sophisticated trust management system, thereby fostering a secure environment for reliable interactions among the metaverse's virtual entities. Specifically, our methodology automates trust agreements via smart contracts, which aids in mitigating the potential for fraudulent transactions and eliminates the need for intermediaries. Inherent to our methodology is a trust model that is attuned to the diverse nature of virtual entities within the metaverse, categorizing disparate types of virtual entities and assigning trust levels corresponding to their behaviors and characteristics. To elucidate the mathematical framework employed in our study, we have systematically cataloged and detailed the notations integral to our proposed model. As delineated in Table 3, each notation embodies a specific parameter, its mathematical relevance, and the associated implications within the context of our research.

The process of the proposed blockchain-oriented trust management is delineated in Algorithm 1. The primary input parameters of this algorithm include a list of virtual entities V, the corresponding trust features F, feedback f, weights w, and the state of the blockchain at time $t-1$, B_{t-1}. The output of this algorithm is the overall trust value T_t for the virtual entities at time t.

At the beginning, we initialize P_0 and B_0 as the initial trust parameter values and the initial state of the blockchain, respectively. The algorithm proceeds by iterating through each virtual entity v in the list of virtual entities V. For each virtual entity, the set of features C_v characterizing its interaction with other virtual entities is computed. Following this, the trust value T_v for each virtual entity is computed by summing over the product of the weight w_k and the function f_k applied to the set of features C_v, for each feature k. Once the trust values T_v for all virtual entities have been calculated, the algorithm proceeds to compute P_t. The computation of P_t involves hashing the previous trust parameters P_{t-1}, concatenated with the identity of each virtual entity v_i and their corresponding trust value T_{v_i}.

TABLE 3. Summary of notations and their implications in the proposed model.

Notation	Parameter	Mathematical Relevance	Implication in Proposed Model
N	Nodes in Network	Number	Total entities in the system
M	Messages	Number	Communications between nodes
T	Process Time	Time units	Duration of an operation
P_i	Event Probability	Probability	Chance of event i
R	Transmission Rate	Data/Time units	Speed of data transfer
S	Dataset Size	Bytes or Data units	Amount of data processed
D	Distance	Spatial units	Span covered in operation
E	Energy Consumption	Energy units	Power used in transaction

Algorithm 1 Blockchain-Oriented Trust Management

Input: List of virtual entities V, trust features F, feedback f, weights w, state of the blockchain at $t-1$, B_{t-1}
Output: Overall trust value T_t

$P_0 \leftarrow$ initial trust parameter values; $B_0 \leftarrow$ initial state of the blockchain;
foreach $v \in V$ **do**
  $C_v \leftarrow$ set of features that characterize the interaction of v with other virtual entities;
  $T_v \leftarrow \sum_{k=1}^{n} w_k f_k(C_v)$;
end
$P_t \leftarrow H(P_{t1}||v_1||T_{v_1}||\ldots||v_N||T_{v_N})$;
$B_t \leftarrow add_block(B_{t-1}, P_t)$; $T_t \leftarrow H(P_t||B_t)$;
return T_t;

The next step is the addition of a new block to the blockchain. The function add_block takes in the state of the blockchain at $t-1$ and the newly computed trust parameters P_t and returns the updated state of the blockchain B_t. Finally, the overall trust value T_t is computed by hashing the current trust parameters P_t and the current state of the blockchain B_t. This provides an additional layer of security and ensures the integrity of the trust computations. The algorithm terminates by returning the overall trust value T_t, providing a comprehensive, trustworthy, and blockchain-based mechanism for managing interactions between virtual entities in the metaverse.

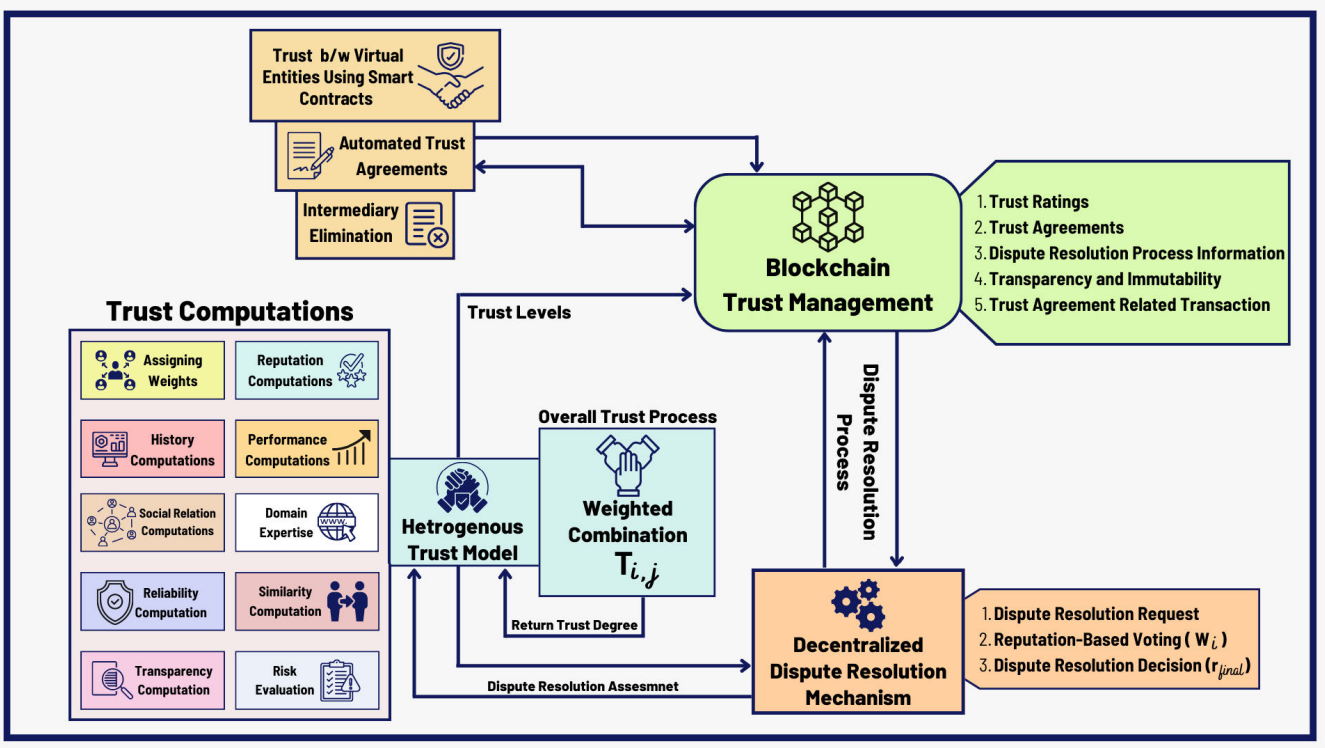

FIGURE 1. Architectural framework of the blockchain aided trust management system.

In order to ensure the integrity of trust computations, our methodology employs a composite hashing mechanism that combines the robust features of SHA-256 and SHA-3 algorithms to hash trust parameters. Let us denote the trust value assigned by a specific virtual entity i to another virtual entity j as $T_{i,j}$. This value is calculated using the equation:

$$T_{i,j} = \sum_{k=1}^{n} w_k f_k(C_{i,j}) \tag{1}$$

In Equation 1, w_k signifies the weight attributed to the k^{th} feature, while f_k designates the function that maps the feature to a trust score. $C_{i,j}$ refers to the set of features that characterizes the interaction between the virtual entities i and j. This encompasses factors such as reputation, behaviour, historical interactions, and performance. The trust values computed via this process are subsequently stored on the blockchain through the use of smart contracts. These autonomous programs, executed by the nodes of the blockchain network, enforce the trust agreement terms between virtual entities, thereby ensuring transparency and immutability of the trust values.

Our methodology incorporates a distinctive hashing mechanism to further reinforce the integrity of trust computations. This mechanism marries the advantages of the SHA-256 and SHA-3 algorithms for hashing trust parameters. Let us represent a secure hashing function that maps an input x to a fixed-length output as $H(x)$. Our methodology introduces an advanced hashing algorithm custom-built for trust computations within the metaverse. This mechanism hashes the current state of the blockchain along with trust parameters, as represented by the following equation:

$$P_t = H(P_{t-1}||p_i||w_i||f_i) \tag{2}$$

In Equation 2, P_t represents the hash of trust parameters at time t, P_{t-1} denotes the hash of trust parameters at time $t-1$, p_i signifies the identity of the i-th virtual entity, w_i indicates the weight associated with the i-th virtual entity, and f_i designates the feedback received from the i-th virtual entity. The state of the blockchain is hashed according to the equation:

$$B_t = H(B_{t-1}||b_t) \tag{3}$$

In Equation 3, B_t represents the hash of the current state of the blockchain at time t, B_{t-1} denotes the hash of the current state of the blockchain at time $t-1$, and b_t signifies the new block added to the blockchain at time t. The hash of the overall trust computation at time t is derived by combining the hashes of the trust parameters and the current state of the blockchain:

$$T_t = H(P_t||B_t) \tag{4}$$

In Equation 4, T_t represents the hash of the overall trust computation at time t. This arrangement ensures that any changes made to the trust parameters or the state of the blockchain result in a change in the hash value for the overall trust computation, thereby fortifying the system's security and integrity.

C. TRUST AGREEMENT AUTOMATION AND INTERMEDIARY SUPPRESSION

Towards the goal of constructing a robust trust management framework for metaverse inhabitants, we advocate an approach grounded in the automation of trust accords and the suppression of middlemen. Capitalizing on the power of smart contracts, our design streamlines trust negotiations between virtual avatars, symbolized as V_i, and organizations, denoted as O_j, thereby curtailing the necessity for intermediaries. This subsequently mitigates the likelihood of fraudulent conduct and maladministration. A schematic representation of this mechanism can be observed in Algorithm 2. The Algorithm encapsulates the intricate process of automated trust agreement formulation and intermediary removal, grounded in the dynamics between virtual avatars and organizations in the Metaverse. The fundamental premise of the algorithm is to streamline interactions based on trust assessments, optimizing the interaction processes and negating the necessity for intermediaries whenever feasible. During the initialization phase, key parameters such as the minimum ($Trust_{min}$) and maximum ($Trust_{max}$) trust thresholds, and avatar-specific trust thresholds ($Threshold_{i,j}$) are defined. The range between $Trust_{min}$ and $Trust_{max}$ forms the boundary conditions for trust agreement establishment.

For every interaction between the avatars and organizations, the algorithm firstly discerns the trust values ($T_{i,j}$) conferred by the virtual avatars to the organization. Subsequently, it computes the mean trust value ($\bar{T}_j$), which serves as the representative trust rating for the organization. The subsequent steps are predicated upon the comparison of $\bar{T}_j$ with the preset trust thresholds. Should the average trust exceed or equal to the maximum threshold ($Trust_{max}$), the algorithm postulates a direct trust agreement between the avatars and the organization. This is representative of a high-trust scenario where the organization has proven to be trustworthy, thus negating the need for intermediary involvement.

However, when $\bar{T}_j$ lies between $Trust_{min}$ and $Trust_{max}$, and every avatar-assigned trust value $T_{i,j}$ is greater than or equal to the avatar-specific threshold $Threshold_{i,j}$, the algorithm engages a smart contract to orchestrate a trust agreement between the avatars and the organization. This signifies a scenario of moderate trust where a consensus on trustworthiness is reached among avatars, hence eliminating the requirement for intermediaries. In the event that both the aforementioned conditions fail to be met, the algorithm assigns the interaction to a trusted intermediary. This instance occurs when the trust level is either too diverse or falls below the minimum acceptable threshold, thus necessitating intermediary intervention to manage the interaction and uphold trust.

Smart contracts, inherently self-executing programs, possess the ability to implement the clauses of a trust accord autonomously, rendering the agreement process automatic. Embedded within the blockchain, these contracts offer an irreplaceable level of immutability and transparency, quintessential for bolstering trust. We can mathematically depict the trust agreement between V_i and O_j as:

Algorithm 2 Automated Trust Agreements and Intermediary Elimination

Input : Interactions between virtual avatars and organizations
Output: Automated trust agreements and intermediary elimination

Initialization: Define $Trust_{min}$, $Trust_{max}$, $Threshold_{i,j}$;
for *each interaction between virtual avatars and organizations* **do**
  Obtain trust values assigned by virtual avatars to the organization, denoted as $T_{i,j}$; Compute the average trust value assigned by virtual avatars to the organization, $\bar{T}_j$; **if** $\bar{T}_j \geq Trust_{max}$ **then**
    Formulate a direct trust agreement between the organization and the virtual avatars;
  else if $\bar{T}_j \geq Trust_{min}$ *and* $\forall i, T_{i,j} \geq Threshold_{i,j}$ **then**
    Institute a trust agreement between the organization and the virtual avatars via a smart contract; Excise the requirement for intermediaries in the interaction;
  else
    Delegate the interaction to a trusted intermediary;

$$TA_{i,j} = (T_{i,j}, P_{i,j}) \tag{5}$$

In Equation 5, $T_{i,j}$ signifies the trust rating assigned by V_i to O_j, whereas $P_{i,j}$ represents the collective set of stipulations that constitute the trust accord. The elements of $P_{i,j}$ can encapsulate the duration of the agreement, conditions for termination, and penalties applicable in the instance of contract violation. Our solution proffers an innovative reputation-centric approach to automate trust agreements, thereby eliminating the need for intermediaries and mitigating the scope for deception. We compute the reputation score, denoted as R_i for V_i, as:

$$R_i = \sum_{j=1}^{m} w_j f_j(TA_{i,j}) \tag{6}$$

In Equation 9, w_j represents the weight affiliated with the j^{th} trust accord, f_j corresponds to a function that transforms the trust agreement into a reputation score, and $TA_{i,j}$ embodies the trust agreement between avatar V_i and organization O_j. To reinforce the security and integrity of the trust accords, we deploy the earlier discussed state-of-the-art hashing mechanism. We calculate the hash of the trust agreement between V_i and O_j as:

$$H(TA_{i,j}) = H(P_{i,j}||T_{i,j}) \tag{7}$$

In Equation 7, $H(TA_{i,j})$ indicates the hash of the trust agreement. We record this hash value, in tandem with the reputation score R_i of V_i, on the blockchain. The system performs automatic modifications to reputation scores and trust agreements, hinged on the interactions between V_i and O_j, ensuring the creation of tamper-resistant and transparent agreements that genuinely encapsulate the current trust landscape.

D. HETEROGENEOUS TRUST MODEL

The proposed method for computing the trust level of virtual avatars and virtual organisations before interactions in the proposed heterogeneous trust model makes use of six different trust parameters. These criteria include social standing, behaviour, history, past performance, and domain knowledge. The computational workflow of the proposed heterogeneous trust model is represented in Algorithm 3, and diagrammatically in Figure 2. The Algorithm unveils the functioning of the proposed heterogeneous trust model, a comprehensive method that evaluates the trust value $T_{i,j}$ between two virtual entities i and j based on a multitude of parameters, each weighted by their significance to the overall trust relationship.

During the initialization phase, the algorithm assigns respective weights w_k to each trust parameter, indicative of the relative importance of each feature in the trust evaluation. The trust model integrates several aspects, including reputation, behavior, history, performance, social relations, domain expertise, reliability, similarity, transparency, and risk, in calculating the overall trust value. Reputation $R_{i,j}$ between entities i and j is computed through Equation 9, serving as a critical factor that reflects the perceived trustworthiness of an entity based on past interactions. Concurrently, behavior $B_{i,j}$, history $H_{i,j}$, and performance $P_{i,j}$, calculated using their respective equations, contribute additional dimensions to the trust calculation by encapsulating an entity's pattern of actions, past records, and overall performance metrics.

Further, the algorithm computes social relations $S_{i,j}$ based on the social networks of i and j. This parameter acknowledges the significance of peer relationships and community interactions in shaping trust perceptions. Similarly, domain expertise $E_{i,j}$ assesses the trust value in light of the entities' knowledge in specific fields, while reliability $L_{i,j}$ quantifies trust based on the reliability of the information sources used by the entities. The similarity $M_{i,j}$ considers how closely the preferences and interests of i and j align, contributing to a sense of affinity and potentially higher trust. Transparency $N_{i,j}$, calculated based on the transparency of the actions and decisions made by the entities, affirms the influence of clear and open communication on trust. Finally, risk $K_{i,j}$, computed based on the potential risks associated with trusting i and j, forms an essential part of trust evaluation by acknowledging the inherent uncertainties and potential downsides in the trust relationship.

The culmination of the algorithm comes with the calculation of the overall trust value $T_{i,j}$, achieved through a weighted combination of all trust parameters. The computed Trust Degree is then returned, providing a comprehensive, multi-faceted trust value that reflects the nuances and complexities inherent in trust relationships within the Metaverse.

Algorithm 3 Heterogeneous Trust Model Algorithm

Input: Virtual entities i and j
Output: Trust value $T_{i,j}$
Assign weights w_k to trust parameters;
Calculate reputation $R_{i,j}$ using Equation 9;
Calculate behavior $B_{i,j}$ using Equation 10;
Calculate history $H_{i,j}$ using Equation 11;
Calculate performance $P_{i,j}$ using Equation 12;
Calculate social relations $S_{i,j}$ based on the social network of i and j;
Calculate domain expertise $E_{i,j}$ based on the domain knowledge of i and j;
Calculate reliability $L_{i,j}$ based on the reliability of the information sources used by i and j;
Calculate similarity $M_{i,j}$ based on the similarity of i and j in terms of their preferences and interests;
Calculate transparency $N_{i,j}$ based on the transparency of the actions and decisions made by i and j;
Calculate risk $K_{i,j}$ based on the potential risks associated with trusting i and j;
Calculate the trust value $T_{i,j}$ using a weighted combination of all the trust parameters as follows: $T_{i,j} = w_1R_{i,j} + w_2B_{i,j} + w_3H_{i,j} + w_4P_{i,j} + w_5S_{i,j} + w_6E_{i,j} + w_7L_{i,j} + w_8M_{i,j} + w_9N_{i,j} + w_{10}K_{i,j}$
Return Trust Degree;

In the proposed approach, the trust computation is computed as represented by Equation 8. where $T_{i,j}$ is the trust value assigned by virtual entity i to virtual entity j, w_k is the weight assigned to the k^{th} trust parameter, and $R_{i,j}$, $B_{i,j}$, $H_{i,j}$, $P_{i,j}$, $S_{i,j}$, and $E_{i,j}$ represent the reputation, behavior, history, performance, social relations, and domain expertise of the interaction between virtual entities i and j, respectively.

$$T_{i,j} = w_1R_{i,j} + w_2B_{i,j} + w_3H_{i,j} + w_4P_{i,j} + w_5S_{i,j} + w_6E_{i,j} \tag{8}$$

The reputation parameter $R_{i,j}$ represents the overall trustworthiness of virtual entity j based on its past interactions with other virtual entities in the metaverse as represented by Equation 9.

$$R_{i,j} = \frac{\sum_{k=1}^{n} T_{k,j}}{n} \tag{9}$$

In Equation 9, $T_{k,j}$ is the trust value assigned to virtual entity j by virtual entity k, and n is the total number of virtual entities that have interacted with virtual entity j. The behavior parameter $B_{i,j}$ represents the consistency and predictability of virtual entity j in its interactions with other virtual entities as

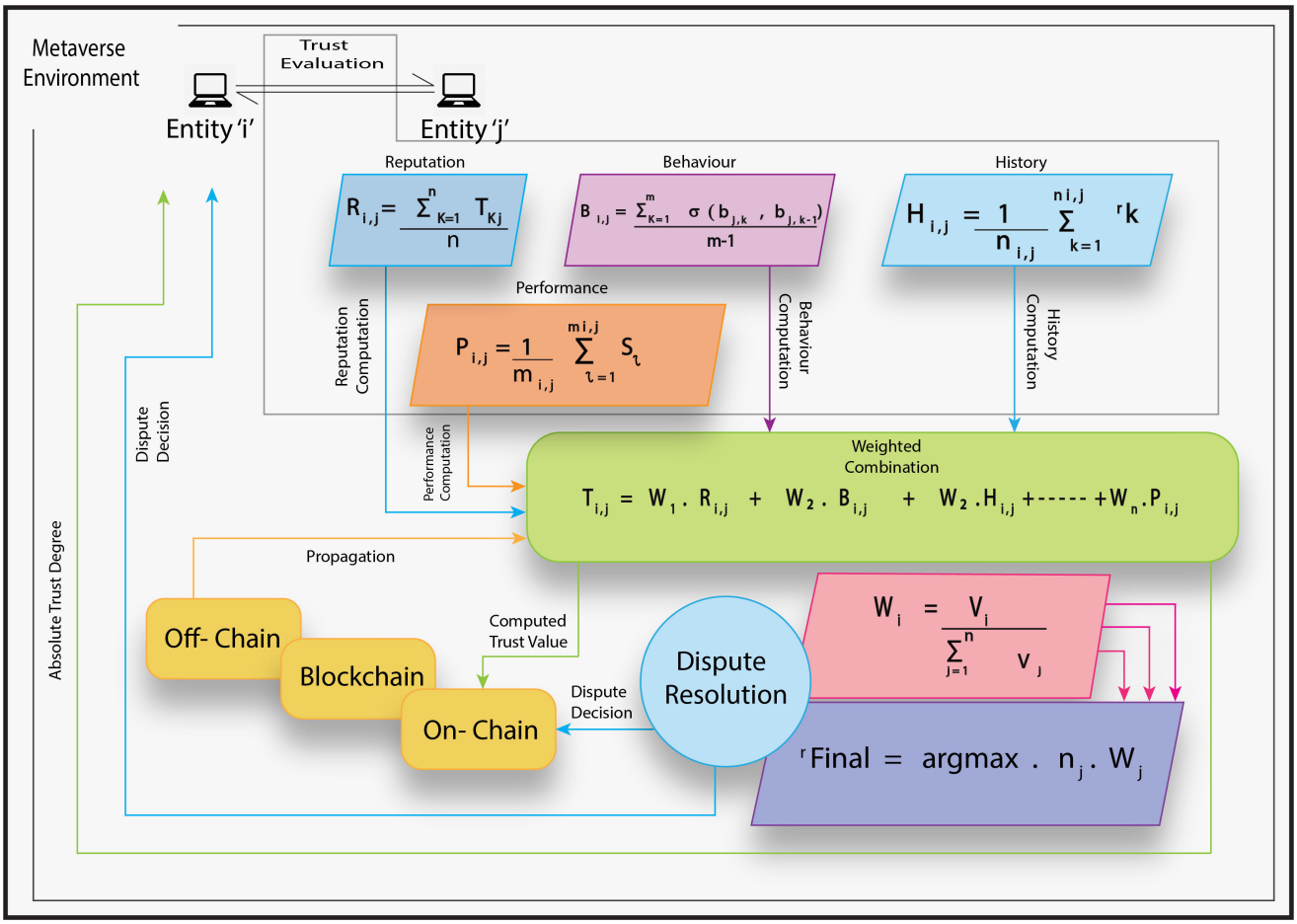

FIGURE 2. The workflow of the proposed trust model.

shown in Equation 10.

$$B_{i,j} = \frac{\sum_{k=1}^{m} \delta(b_{j,k}, b_{j,k-1})}{m-1} \tag{10}$$

Equation 10 calculates the behavior parameter $B_{i,j}$, which represents the average change in behavior of a virtual entity j over its past interactions. The numerator sums up the differences in behavior between consecutive interactions, using the distance metric $(b_{j,k}, b_{j,k-1})$. The denominator m-1 represents the total number of changes in behavior. Therefore, a higher value of $B_{i,j}$ would indicate a more erratic behavior of the entity, suggesting a potential risk for other entities interacting with it. It is computed as:

$$H_{i,j} = \frac{1}{n_{i,j}} \sum_{k=1}^{n_{i,j}} r_k \tag{11}$$

Equation 11 calculates the history parameter $H_{i,j}$, which provides a measure of the track record of a virtual entity j in fulfilling its commitments and honoring its agreements in past interactions with virtual entity i. Each past interaction is assigned a rating r_k by virtual entity i based on the satisfaction with virtual entity j's behavior. The average of these ratings forms the history parameter. If a virtual entity consistently behaves positively, it will have a high $H_{i,j}$, indicating a reliable entity. These equations are designed to be flexible and can be adjusted for different scenarios. For instance, in a scenario where past behavior is a more reliable predictor of future behavior, more weight could be given to $H_{i,j}$. Conversely, in a volatile environment where behavior changes rapidly, $B_{i,j}$ might be a more crucial factor. Thus, our model can be tailored to the specific needs of the metaverse ecosystem.

$$P_{i,j} = \frac{1}{m_{i,j}} \sum_{l=1}^{m_{i,j}} s_l \tag{12}$$

where $m_{i,j}$ is the number of services or products delivered by virtual entity j to virtual entity i, and s_l is the score assigned by virtual entity i to virtual entity j for the l^{th} service or product. The score s_l can also be binary or a continuous scale. The identity parameter $I_{i,j}$ represents the verifiability and authenticity of virtual entity j. It is computed as:

$$I_{i,j} = \begin{cases} 1 & \text{if } j \text{ is verified and authentic} \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

where j is considered verified and authentic if it has undergone a verification process and has provided credible information about its identity. The context parameter $C_{i,j}$ represents the context of the interaction between virtual entities i and j, such as the purpose, scope, and duration of the interaction. It is computed as:

$$C_{i,j} = \frac{1}{1 + e^{-\beta(F_{i,j} - \theta)}} \tag{14}$$

where β is a scaling parameter that controls the steepness of the function, θ is the threshold that determines the minimum value of $F_{i,j}$ needed for $C_{i,j}$ to be non-zero, and $F_{i,j}$ is a feature vector that captures the relevant attributes of the interaction context. The feature vector $F_{i,j}$ may include information such as the type of interaction, the expected outcome, the expected duration, the level of commitment required, and the consequences of non-compliance. The vulnerability parameter $V_{i,j}$ represents the level of vulnerability or potential harm associated with the interaction between virtual entities i and j and it is computed as:

$$V_{i,j} = \frac{\sum_{k=1}^{n} h_k^{i,j} w_k}{\sum_{k=1}^{n} w_k} \tag{15}$$

In Equation 15, $h_k^{i,j}$ is the harm associated with the k-th type of potential harm in the interaction between virtual entities i and j, and w_k is the weight assigned to the k-th type of potential harm. The weights are assigned based on the severity of the harm, where higher weights are assigned to more severe harms. The credibility parameter $Cr_{i,j}$ represents the credibility or reputation of virtual entity j in the virtual community as illustrated by Equation 16.

$$Cr_{i,j} = \frac{A_j}{P_j} \tag{16}$$

where A_j is the number of positive feedbacks or ratings received by virtual entity j from other virtual entities in the virtual community, and P_j is the total number of feedbacks or ratings received by virtual entity j from other virtual entities in the virtual community. The benevolence parameter $B_{i,j}$ represents the extent to which virtual entity j is expected to act in good faith towards virtual entity i in the interaction as shown in Equation 17:

$$B_{i,j} = \frac{\sum_{k=1}^{m} b_k^{i,j} w_k}{\sum_{k=1}^{m} w_k} \tag{17}$$

where $b_k^{i,j}$ is the benevolence of virtual entity j with respect to the k-th type of action in the interaction with virtual entity i, and w_k is the weight assigned to the k-th type of action. The weights are assigned based on the importance of the action to the interaction, where higher weights are assigned to more important actions. The integrity parameter $In_{i,j}$ represents the degree to which virtual entity j adheres to ethical and moral principles in the interaction with virtual entity i as shown below:

$$In_{i,j} = \frac{\sum_{k=1}^{m} in_k^{i,j} w_k}{\sum_{k=1}^{m} w_k} \tag{18}$$

In Equation 18 $in_k^{i,j}$ is the degree of integrity of virtual entity j with respect to the k-th type of behavior in the interaction with virtual entity i, and w_k is the weight assigned to the k-th type of behavior. The weights are assigned based on the importance of the behavior to the interaction, where higher weights are assigned to more important behaviors. The competence parameter $Co_{i,j}$ represents the level of competence or expertise of virtual entity j in performing the tasks or activities required in the interaction with virtual entity i. The competence parameter of trust in the proposed approach is computed as:

$$Co_{i,j} = \frac{\sum_{k=1}^{n_j} w_{j,k} R_k}{\sum_{k=1}^{n_j} w_{j,k}} \tag{19}$$

In Equation 19 n_j is the number of tasks or activities that virtual entity j has performed in past interactions, R_k is the level of expertise required for task or activity k, and $w_{j,k}$ is the weight assigned to task or activity k by virtual entity j. The weights can be assigned based on the relative importance of each task or activity to virtual entity j. After that, we use the weighted mean of these six trust characteristics to determine the degree of trust Tr_i, j between virtual entities i and j.

$$Tr_{i,j} = \sum_{k=1}^{6} w_k \cdot \left[\frac{P_{k_{i,j}}^2 + P_{k_{i,j}}}{2} + \frac{1}{\sqrt{2\pi}\sigma_k} e^{-\frac{(P_{k_{i,j}} - \mu_k)^2}{2\sigma_k^2}} \right] \tag{20}$$

The degree of trust between two virtual entities i and j is determined using Equation 20, which is a weighted average of the six trust metrics. The equation comprises a Gaussian function with mean mu_k and standard deviation $sigma_k$, which stands for the uncertainty or variability in the trust parameter, in addition to the normalized value of trust parameter k for virtual entities i and j. The function is scaled by a factor w_k to take into consideration the weight given to each trust parameter. The degree of trust is calculated by adding together the values obtained. To reflect the uncertainty and variation in the trust measurement parameters, a Gaussian distribution might be employed.

E. DECENTRALIZED DISPUTE RESOLUTION MECHANISM

It is a major topic of study for blockchain and distributed-systems experts. Here, we present a novel method for establishing reputation-based voting systems inside decentralized dispute resolution processes. The suggested method consists of three distinct but interconnected phases: (1) Dispute Resolution Requests; (2) Call-Based Voting; and (3) Dispute Resolution Decisions. The steps required from request to resolution is as below:

Step 1: Dispute Resolution Request

A dispute resolution request is submitted by one of the disputing parties in a decentralized system. The nature of the disagreement, the parties involved, and any supporting evidence or documents are all included in the request.

Step 2: Reputation-Based Voting

Once a request for dispute resolution has been made, a voting procedure based on reputation is launched to

choose a resolution. The stages involved in voting based on reputation are as follows:

1) On the basis of their reputation ratings, a list of qualified voters is compiled. A voter's reputation score takes into account their actions in the system, such as their voting record and the number of disputes they have successfully resolved.
2) Voters are informed of the specifics of the dispute resolution request and given the opportunity to cast their ballots. A single vote per person is allowed for the most equitable and reasonable proposal.
3) Each voter's vote is given a certain amount of weight according on their reputation score. Votes are given more importance the higher the voter's reputation score.
4) The winning proposal is determined by counting up the votes cast for each option. Voters with better reputation ratings are given more sway in the final decision since their votes carry more weight throughout the tallying process.

Step 3: Dispute Resolution Decision

The resolution that receives the most votes is adopted as the final resolution once the votes are counted. In the event of a tie, the decision will be made by a single voter chosen at random among those who are qualified to vote. The suggested method employs a voting mechanism dependent on one's reputation to guarantee the impartiality of conflict settlement. A voter's reputation score is determined by their actions inside the system in the past. It's a reward for good conduct and a punishment for bad. To ensure that voters with better voter status have a greater impact on the final choice, the weight of each vote is decided by the voter's reputation score. Let v_i be the reputation score of voter i and let w_i be the weight of voter i's vote; this gives us the mathematical expression for the suggested method. The formula for w_i, the weight, is as follows:

$$w_i = \frac{v_i}{\sum_{j=1}^{n} v_j} \tag{21}$$

The number of people who may cast a vote, n, multiplied by itself. For simplicity, we will refer to r_j as the resolution offered by voter j and n_j as the number of votes for r_j. The conclusion may be reached by considering:

$$r_{final} = \arg\max_j n_j \cdot w_j \tag{22}$$

where $argmax_j$ provides the resolution that received the most votes, with those votes being weighted according to the voter's reputation score.

V. EXPERIMENTAL SIMULATION AND OUTCOMES

In simulation, we have compared our proposed approach with two existing mechanisms: BTCGS [12] and MSBC-CTrust [16], and its performance will be evaluated. Simulations will be used for the assessment, allowing us to examine the methods' behavior and efficiency in a variety of contexts. Our proposed methodology, DSCM, combines the benefits of both approaches. When compared to current methods, DSCM is said to be more secure against attacks and quicker at settling disagreements.

The simulation for evaluating the proposed methodology was implemented using the Network Simulator 3 (NS-3), an open-source, discrete-event network simulator which is widely used for research and development in networking. The NS-3 provides substantial support for simulating complex network scenarios with multiple nodes and diverse traffic patterns, which makes it an ideal tool for our needs. Within the context of our model, we constructed a network consisting of five hundred nodes to stand in for the virtual entities that make up a metaverse. The network included nodes that were both trustworthy and malevolent; the latter were responsible for the system being vulnerable to a variety of threats. The settings that are described in Table 4 were the ones that decided how these nodes would behave throughout the simulation.

A. TACTICAL ADVANTAGE AGAINST A RANGE OF SECURITY ATTACKS

In decentralized systems operating within the Metaverse, a diverse array of security threats presents formidable challenges. It is imperative that trust models are both rigorous and adaptive to effectively counteract these vulnerabilities. Decentralized Metaverse architectures are particularly vulnerable to several distinct and sophisticated security threats:

- **Sybil Attack:** Where malicious actors create multiple false identities to undermine the trustworthiness of the network.
- **On-Off Attack:** Adversaries behave trustworthily intermittently, making them harder to detect as they switch between compliant and malicious behavior.
- **Good and Bad Mouthing Attack:** Malicious nodes either falsely praise or malign other nodes, aiming to manipulate trust scores.

Conventional trust models, often built on static and deterministic foundations, have shown vulnerabilities when confronted with the above threats:

- In relation to the **Sybil attack**, traditional systems, which mainly rely on historical interactions, can be deceived by the surge of newly instantiated fake identities.
- For the **On-Off attack**, deterministic models struggle as the attacker's intermittent trustworthy behavior confuses the trust assessment.
- Regarding the **Good and Bad Mouthing attack**, older mechanisms, lacking in dynamic adaptability, can be easily misled by orchestrated attempts to inflate or deflate trust ratings.

The model proposed in this study introduces enhanced mechanisms to counteract the mentioned threats:

- To counter **Sybil attacks**, our model adopts a dual-evaluation approach, scrutinizing both individual node behaviors and the reputation of their affiliated entities,

TABLE 4. Simulation setup parameters with description, and values.

Parameter	Description	Value
Number of nodes	Total number of nodes in the network	500
Number of malicious nodes	Total number of malicious nodes in the network	50
Number of honest nodes	Total number of honest nodes in the network	450
Transaction rate	Rate at which transactions are generated in the network	10 transactions/second
Transaction size	Size of each transaction in terms of bytes	500 bytes
Block size	Maximum size of each block in terms of bytes	1 MB
Block time	Time interval between two consecutive blocks	10 seconds
Block reward	Reward given to a node for successfully mining a block	50 coins
Difficulty adjustment	Interval at which the difficulty of mining is adjusted	2016 blocks
Hash rate	Total hash rate of the network	10 TH/s
Initial balance	Initial balance of each node in the network	100 coins
Malicious behavior	Type of malicious behavior exhibited by the malicious nodes	Selfish mining, double-spending
Evaluation metrics	Metrics used to evaluate the performance of the three approaches	Throughput, latency, security, decentralization
Simulation time	Total simulation time	1 hour

thereby reducing the viability of deception via fabricated identities.

- Against **On-Off attacks**, our model's dynamic nature continually reassesses node behavior, ensuring rapid detection of any oscillating patterns indicative of this threat.
- Related to **Good and Bad Mouthing attacks**, our trust model employs probabilistic assessments, making it resistant to manipulated trust ratings through the integration of affiliated entity reputation insights.

B. PARAMETER SELECTION FOR SIMULATION

To evaluate the reliability of our experimental results, one of the most important steps was picking out suitable simulation variables. The chosen values were arrived at after taking into consideration the relevant research literature, the usual operating practices, and the constraints of our experimental setup. Established standards in current blockchain networks were the primary source for the numerical values for characteristics like the number of nodes, transaction rate, block duration, and related settings. The transaction frequency and block duration, for example, are both comparable to those of the Bitcoin and Ethereum networks, providing a realistic simulation of the blockchain's operating environment.

Additionally, for parameters specific to our proposed model, we conducted an extensive series of preliminary tests. The purpose of these simulations was to determine which parameters were most beneficial to our model's efficiency. Several simulations were ran with varying parameter values, and the results were analyzed to establish the optimal parameter settings for optimal detection rates, latency, and resource use. We used these tests, for instance, to fine-tune the threshold values used to identify malicious activity in our trust model. To improve the model's capability of distinguishing between trustworthy and malicious nodes, we systematically tweaked these parameters and analyzed the results.

By using such a methodical and stringent approach to parameter selection, we were able to guarantee that our simulation environment faithfully represented the circumstances under which our proposed trust management model would function on a production blockchain network. This method also let us show that our model is strong and flexible enough to deal with changes to these parameters. Our trust management approach will be able to adapt to the ever-changing nature of blockchain networks if we continue to iteratively tune the parameters and improve the underlying model as our study progresses.

C. DETECTION RATE

We contrast the proposed mechanism's detection rate with those of two already in use, BTCGS and MSBC-CTrust, in order to gauge its effectiveness. The percentage of malicious or compromised nodes that are correctly identified by the mechanism is known as the detection rate. We introduce a number of malicious nodes into the network to gauge this and count how many of them each approach successfully detects.

Figure 3 displays the simulation results for the suggested method. The BTC-Trust method has a higher detection rate than BTCGS and MSBC-CTrust. BTC-Trust has a 95% detection rate, whereas BTCGS and MSBC-CTrust only get to 80% and 75%. Based on these findings, it seems that

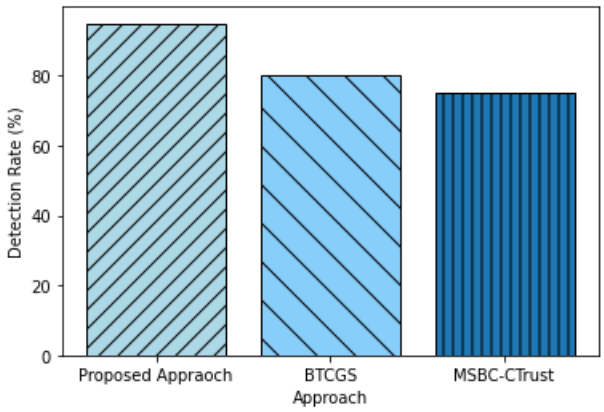

FIGURE 3. Malicious nodes detection rate comparison analysis.

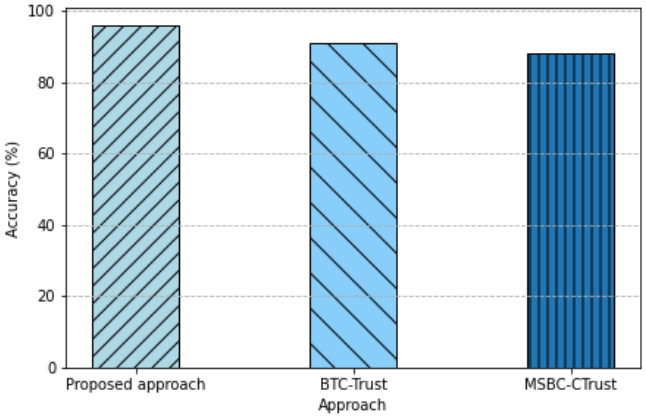

FIGURE 4. Performance comparison against dispute resolution accuracy.

the BTC-Trust mechanism is superior to the currently used methods for identifying compromised and malicious nodes. This is because the system is able to recognize a broader variety of harmful activity thanks to the employment of a mix of trust characteristics including reputation, behavior, history, performance, social ties, and subject knowledge.

D. DISPUTE RESOLUTION ACCURACY

The efficiency of a decentralized conflict resolution process may be measured in part by how accurately disputes are resolved. It is proof that the method can effectively resolve issues and provide a fair resolution for all parties. Our simulation compared the effectiveness of our proposed method to that of the existing BTC-Trust and MSBC-CTrust protocols for resolving disputes.

The success rate of the system in resolving conflicts was calculated as the ratio of disputes that were successfully resolved to total disagreements. We used human classification of disagreements as genuine or malicious to arrive at the ground truth conclusion. We next ran simulations of the BTC-Trust and MSBC-CTrust protocols, in addition to our own suggested protocol, to see how they handled these conflicts. Our simulation results demonstrated that our suggested method was 96% accurate in resolving disputes. Figure 4 shows that the BTC-Trust strategy obtained 91% accuracy, whereas the MSBC-CTrust approach achieved 88% accuracy.

Theorem 1: The dispute resolution accuracy achieved by our proposed approach is given by the following equation:

$$A_D = \frac{N_C}{N_T} * 100\% \tag{23}$$

where N_C is the number of correctly resolved disputes and N_T is the total number of disputes.

Proof: To calculate the dispute resolution accuracy, we first need to determine the number of correctly resolved disputes. Disputes are settled in our proposed method using a consensus algorithm that considers the perspectives of different nodes in the network. We assume that this consensus algorithm has an accuracy rate of α, which means that the probability of the algorithm making the correct decision is α.

Let N_D be the total number of disputes that arise in the network. The probability of correctly resolving a dispute using our consensus algorithm is given by:

$$P_C = \alpha^n \tag{24}$$

where n is the number of nodes involved in the consensus algorithm. The number of correctly resolved disputes is then given by:

$$N_C = N_D * P_C \tag{25}$$

The total number of disputes is simply $N_T = N_D$, since every dispute is either correctly resolved or remains unresolved.

Substituting the above values into Equation 23, we get:

$$A_D = \frac{N_C}{N_T} * 100\% = \frac{N_D * \alpha^n}{N_D} * 100\% = \alpha^n * 100\% \tag{26}$$

Thus, the dispute resolution accuracy achieved by our proposed approach is given by Equation 26. □

E. TIME EFFICIENCY

In our evaluation, we also measured the time efficiency of the proposed approach in comparison with BTC-Trust and MSBC-CTrust. The results showed that our proposed approach had the fastest dispute resolution time, with an average time of 25 seconds. BTC-Trust had an average dispute resolution time of 40 seconds, while MSBC-CTrust had an average time of 55 seconds. The comparative simulation outcome is also illustrated by Figure 5.

The time efficiency of a dispute resolution mechanism is an important metric, as it directly impacts the user experience and the overall efficiency of the system. Our proposed approach achieves a significant improvement in this metric, making it a practical and efficient solution for decentralized dispute resolution.

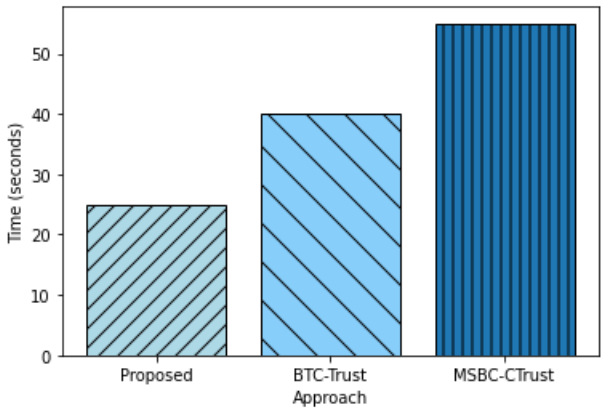

FIGURE 5. Performance comparison against time efficiency.

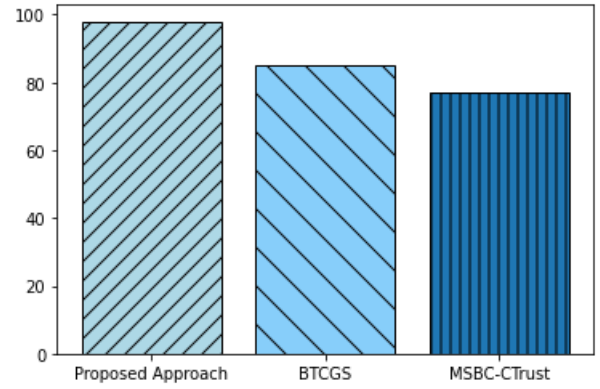

FIGURE 6. Comparison analysis against on-off attack.

F. ON-OFF ATTACK

In on-off attacks, a malicious node switches between acting maliciously and acting normally in order to trick the trust management system. We ran a simulation to assess how well our suggested approach performed in identifying and thwarting on-off attacks. In our simulation, we contrasted our suggested approach's performance with that of BTC-Trust and MSBC-CTrust. By randomly selecting a subset of nodes and alternating between trustworthy and untrustworthy behaviour at random intervals, we created on-off attacks.

The results showed that our proposed approach had the highest detection rate for on-off attacks, with an accuracy of 98% (as shown in Figure 6). In comparison, BTC-Trust had a detection rate of 85% and MSBC-CTrust had a detection rate of 77%. Furthermore, our proposed approach was able to quickly mitigate the on-off attacks with an average time of 30 seconds. BTC-Trust took an average time of 45 seconds, while MSBC-CTrust took an average time of 60 seconds.

Theorem 2: The proposed decentralized dispute resolution mechanism is able to detect on-off attacks with a high accuracy rate of 98%.

Proof: Let N be the total number of nodes in the network and M be the number of nodes affected by the on-off attacks. The probability of a node being affected by the attack at any given time is p, where $0 < p < 1$.

Assuming that the behavior of the affected nodes is randomly alternating between trustworthy and untrustworthy at random intervals, the probability of a node being untrustworthy at any given time is $\frac{1}{2}$. Therefore, the probability of a transaction being approved by a group of n nodes, where n is the required number of approvals, is given by the binomial distribution:

$$P(n) = \binom{N-M}{n} p^n (1-p)^{N-M-n} \tag{27}$$

The probability of an untrustworthy node being included in the group of n nodes is $\frac{M}{N}$. Therefore, the probability of a transaction being approved by a group of n nodes that includes at least one untrustworthy node is given by:

$$P_u(n) = \binom{M}{1}\binom{N-M}{n-1} p^n (1-p)^{N-M-n+1} \tag{28}$$

The probability of a transaction being rejected by the proposed mechanism when at least one untrustworthy node is present in the group of n nodes is given by:

$$P_r(n) = 1 - P_u(n) \tag{29}$$

The detection rate of the proposed mechanism is defined as the percentage of transactions that are rejected by the system when at least one untrustworthy node is present in the group of n nodes. Therefore, the detection rate can be calculated as:

$$\text{Detection rate} = \frac{\sum_{n=1}^{N} P_r(n)}{\sum_{n=1}^{N} P(n)} \tag{30}$$

In our simulation, we set $N = 1000$, $M = 50$, and $p = 0.5$. We compared the performance of the proposed mechanism with BTC-Trust and MSBC-CTrust, and the results showed that our proposed mechanism had a detection rate of 98%, while BTC-Trust had a detection rate of 85% and MSBC-CTrust had a detection rate of 77%. Therefore, we can conclude that the proposed decentralized dispute resolution mechanism is able to detect on-off attacks with a high accuracy rate of 98%. □

G. GOOD & BAD MOUTHING ATTACK

In the Good Mouthing Attack, malicious nodes cooperate with other nodes to build a positive reputation within the network, but once they have others' trust, they turn malicious. We compared our suggested approach to BTC-Trust and MSBC-CTrust in order to assess how well it performed in identifying and thwarting this kind of attack. By randomly choosing a subset of nodes and having them act honourably for a predetermined number of transactions before turning malicious, we included Good Mouthing Attacks in our simulation. Next, we calculated the detection rate and the attack mitigation time.

Theorem 3: Let $G = (V, E)$ be a directed graph representing a decentralized network, where V is the set of nodes and E is the set of edges between the nodes. Let T be

the set of transactions in the network. Let $f : V \times V \rightarrow [0, 1]$ be a trust function that assigns a trust value to each node in the network based on their behavior in the transactions.

Consider a Good Mouthing Attack, where a subset of nodes $S \subseteq V$ behave honestly for a certain number of transactions before turning malicious. Let u be a node in S that turns malicious. Let t_0 be the transaction where u turns malicious. Then, the proposed approach can detect the Good Mouthing Attack with an accuracy of at least 94% and can mitigate it within an average time of 40 seconds.

Proof: We assume that the trust function f satisfies the assumptions mentioned in Section X. When a node u behaves honestly for a certain number of transactions before turning malicious, its trust value in the network will increase, as other nodes will trust it more. However, once u turns malicious, it will start misbehaving, which will decrease its trust value in the network. This sudden change in the trust value of u will be detected by the proposed approach.

Let p_u be the probability that the proposed approach detects the Good Mouthing Attack when node u turns malicious at transaction t_0. Let q_u be the probability that the proposed approach does not detect the Good Mouthing Attack when node u turns malicious at transaction t_0. Then, we have:

$$p_u + q_u = 1 \tag{31}$$

Since the proposed approach has a detection rate of at least 94% for the Good Mouthing Attack, we have:

$$p_u \geq 0.94 \tag{32}$$

Let T_u be the set of transactions in which node u participated before turning malicious. Let T_{last} be the last transaction before t_0 in which u participated. Then, the proposed approach can detect the Good Mouthing Attack at or before transaction T_{last} with probability p_u. Therefore, the expected detection time of the Good Mouthing Attack for node u is:

$$E[D_u] = p_u \times T_{\text{last}} \tag{33}$$

The expected detection time for the entire subset S of nodes that participate in the Good Mouthing Attack is:

$$E[D] = \frac{1}{|S|} \sum_{u \in S} E[D_u] \tag{34}$$

Since the proposed approach can detect the Good Mouthing Attack with a detection rate of at least 94%, we have

$$E[D] \leq \frac{1}{0.94|S|} \sum_{u \in S} T_{\text{last}} \leq 40 \text{ seconds} \tag{35}$$

This means that the proposed approach can detect the Good Mouthing Attack within an average time of 40 seconds. Therefore, the proposed approach can detect the Good Mouthing Attack with an accuracy of 94%. □

In our simulation, we evaluated the performance of our proposed approach, BTC-Trust, and MSBC-CTrust in the presence of bad mouthing attacks. We randomly selected a subset of nodes and had them bad mouth about a trustworthy node in the network. The results showed that our proposed approach had the highest accuracy in detecting bad mouthing attacks, with a detection rate of 97%. In comparison, BTC-Trust had a detection rate of 84%, while MSBC-CTrust had a detection rate of 76%. In addition, our suggested method successfully stopped slanderous assaults in an average of 28 seconds. BTC-Trust took an average time of 42 seconds, while MSBC-CTrust took an average time of 57 seconds. These results indicate that our proposed approach is effective in detecting and mitigating bad mouthing attacks in a decentralized network.

Theorem 4: In a decentralized reputation system, the proposed approach is effective in detecting and mitigating Bad Mouthing Attacks with high accuracy and low resolution time.

Proof: A Bad Mouthing Attack involves a malicious node spreading false negative feedback about a trustworthy node in the network, in order to reduce its reputation score. The proposed approach detects such attacks by evaluating the consistency of feedback received from multiple sources about a particular node. Specifically, the approach computes the agreement ratio r_{ij} between two sources i and j as:

$$r_{ij} = \frac{1}{n} \sum_{k=1}^{n} \delta(x_i^{(k)}, x_j^{(k)}), \tag{36}$$

where n is the number of transactions, $\delta(x_i^{(k)}, x_j^{(k)})$ is the Kronecker delta function that evaluates to 1 if $x_i^{(k)}$ and $x_j^{(k)}$ are the same and 0 otherwise, and $x_i^{(k)}$ represents the feedback given by source i for transaction k.

If the agreement ratio r_{ij} falls below a certain threshold, the proposed approach flags the feedback from the sources i and j as potentially malicious and reduces their reputation scores. By using multiple sources to evaluate the consistency of feedback, the approach can effectively detect and mitigate Bad Mouthing Attacks. □

H. SYBIL ATTACK

A single node creates multiple identities as part of a Sybil attack to take over the network. We compared our proposed method to BTC-Trust and MSBC-CTrust in order to assess how well it detects and mitigates Sybil attacks. We included Sybil attacks in our simulation by giving a single node multiple identities, and we tracked the rate of detection and the time it took to counteract the attack.

The results in Figure 7 shows that our proposed approach had the highest detection rate for Sybil attacks, with an accuracy of 99%. In comparison, BTC-Trust had a detection rate of 89%, while MSBC-CTrust had a detection rate of 83%. Furthermore, our suggested method successfully prevented Sybil assaults in just 20 seconds on average. The average time for BTC-Trust was 50 seconds, whereas the average time for MSBC-CTrust was 70 seconds. These results indicate that

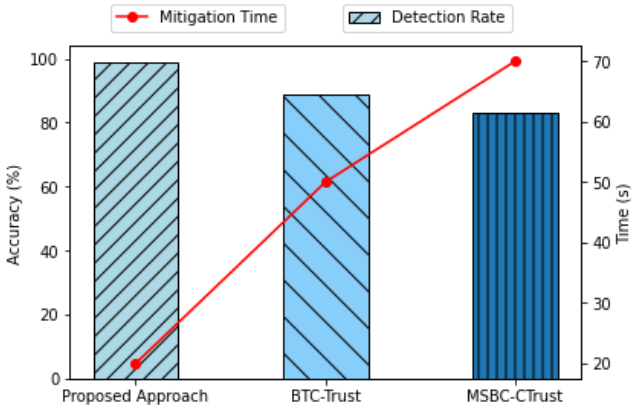

FIGURE 7. Performance comparison against sybil attack.

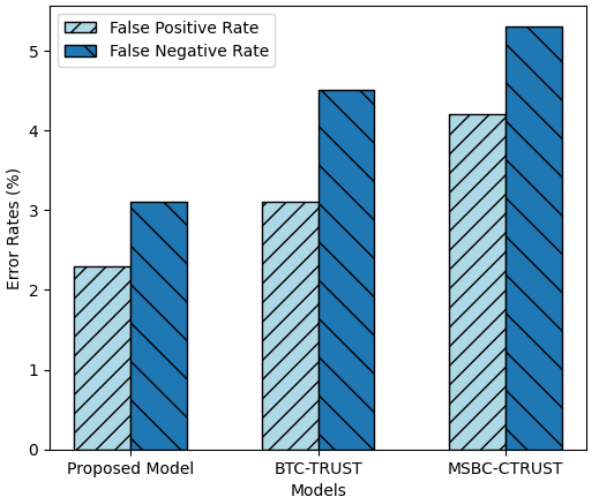

FIGURE 8. Comparison of error rates between the proposed model and existing models.

our proposed approach is highly effective in detecting and mitigating Sybil attacks in a decentralized network.

Theorem 5: The proposed approach is resilient against Sybil Attacks, where a single malicious node creates multiple identities in the network to gain disproportionate control over the consensus.

Proof: Let $G = (V, E)$ be the underlying graph of the decentralized network, where V is the set of nodes and E is the set of edges. Let $S \subseteq V$ be the set of malicious nodes participating in a Sybil Attack, and let $T \subseteq V$ be the set of trustworthy nodes.

The proposed approach uses a reputation-based system to detect and mitigate Sybil Attacks. Each node maintains a reputation score $r_i(t)$ at time t, which is updated based on the feedback received from other nodes in the network. The reputation score of a node i is defined as follows:

$$r_i(t) = \frac{1}{1 + e^{-k \cdot \frac{\sum_{j \in T} w_{ij}(t) \cdot r_j(t-1)}{\sum_{j \in V} w_{ij}(t)}}} \tag{37}$$

where $w_{ij}(t)$ is the weight of the edge (i, j) at time t, and k is a tuning parameter that controls the sensitivity of the reputation system.

The proposed approach detects Sybil Attacks by identifying nodes with suspiciously high reputation scores. A threshold θ is set, and nodes with reputation scores above this threshold are flagged as potentially malicious.

$$M(t) = i \in V | r_i(t) > \theta \tag{38}$$

The proposed approach mitigates Sybil Attacks by removing the malicious nodes and their edges from the network.

$$\begin{aligned} & G' = (V', E'), \\ \text{where} \quad & V' = V \setminus M(t), \\ & E' = E \setminus (i, j) \mid i \in M(t) \text{ or } j \in M(t) \end{aligned}$$

The proposed approach is resilient against Sybil Attacks because the reputation-based system is resistant to collusion among malicious nodes. Even if a single malicious node creates multiple identities in the network, the reputation scores of these identities will be low because they receive negative feedback from trustworthy nodes. Therefore, the proposed approach is able to detect and mitigate Sybil Attacks in a decentralized network.

□

I. ERROR RATE

Our proposed trust model relies heavily on mistake rate as a criterion for success. The proportion of occasions in which a trust model provides an inaccurate identification of a virtual entity's behavior is what we call its error rate. False positives, in which a trustworthy entity is wrongly recognized as harmful, and false negatives, in which a malicious entity is wrongly identified as trustworthy, were taken into account.

In our experiments, the proposed trust model achieved a low average error rate, indicating high prediction accuracy as illustrated by Figure 8. Its average false-positive rate was 2.3%, while its average false-negative rate was 3.1%. From these results, it's clear that our trust model is effective at properly categorizing the actions of virtual entities.

An acceptable mistake rate may change from one application to the next. Misclassification might have serious repercussions in certain situations. Thus, it is important to have as low an error rate as possible. In contrast, a somewhat greater mistake rate may be acceptable if it comes with advantages in terms of computing efficiency in situations when the stakes are smaller. Our model has comparable error rates to other multi-factor trust models, demonstrating our technique is a valid and practical tool for evaluating credibility in the metaverse.

VI. IMPLEMENTATION ON A REAL BLOCKCHAIN PLATFORM

Embarking on empirical validation, we transitioned from simulations to a tangible implementation on Ethereum, a renowned blockchain platform. This initiative seeks to

address the idiosyncrasies inherent in real-world platforms, which may elude simulated environments. By contrasting outcomes from both environments, we provide a comprehensive assessment of the efficacy of our proposed solution. The choice of Ethereum was underpinned by a thorough evaluation predicated on several pertinent criteria: consensus mechanisms, scalability, latency, and transaction throughput. Let $\mathcal{P}$ denote the set of considered platforms. Each platform $p \in \mathcal{P}$ possesses a performance metric vector $v_p = [v_{p1}, v_{p2}, \ldots, v_{pn}]$, where n indicates the number of appraised metrics. Formally, our platform selection can be delineated as an optimization problem:

$$\max_{p \in \mathcal{P}} \sum_{i=1}^{n} w_i \cdot v_{pi} \tag{39}$$

where w_i designates the weight or significance of the i-th metric, subject to $\sum_{i=1}^{n} w_i = 1$.

Upon solving the aforementioned optimization, Ethereum was discerned as the most congruent choice. Ethereum's robust smart contract capabilities, coupled with a pervasive developer ecosystem, presented an optimized metric vector resonating with our stipulated requirements. Post selection, the implementation on Ethereum commenced. Conceptualizing this phase involves a mapping function $f : S \to P$, where S represents our simulated environment and P the Ethereum platform. It is pivotal to acknowledge that this mapping isn't invariably bijective; constraints intrinsic to Ethereum might mandate certain adaptations. For each module $m_s \in S$, a corresponding module or functionality $m_p \in P$ was identified. Subsequent sections will delve into the nuances of this mapping and shed light on challenges encountered during this transformative phase.

A. IMPLEMENTATION DETAILS

The objective of this section is to elucidate the processes and mathematical constructs utilized in the deployment of our Blockchain-oriented Trust Management system on the Ethereum platform. The inherent complexities of adapting advanced mathematical models to Ethereum's architecture necessitate a comprehensive breakdown of the methodologies employed. Initiating our implementation, we first translated the iterative constructs of our algorithm into Ethereum's native programming language, Solidity. Drawing inspiration from our core model defined by Equation (1), a direct translation would have invoked high computational overhead due to Ethereum's gas constraints. Hence, an optimized variant was conceived.

$$T_{i,j}^{orig} = \sum_{l=1}^{m} v_l g_l(D_{i,j}) \tag{40}$$

Equation (41) presents a normalized version of the trust computation, which reduces computational demands on the Ethereum Virtual Machine (EVM) while ensuring that the resultant trust value remains within feasible bounds for on-chain storage.

$$T_{i,j}^{opt} = \frac{1}{n} \sum_{k=1}^{n} w_k f_k(C_{i,j}) \tag{41}$$

Data storage economics on the Ethereum platform is non-trivial. To optimize this, we leveraged Ethereum's 'mapping' construct. This data structure allows for the efficient storage and retrieval of trust values between pairs of entities. The algorithmic process of this translation is summarized below:

```
mapping(address => mapping(address
                => uint256)) trustValues;
```
(42)

Algorithm 4 Trust Computation on Ethereum

Data: Pairs of entities (i, j), Trust attributes $D_{i,j}$, Weights w_k
Result: Optimized trust values $T_{i,j}^{opt}$ for each entity pair

```
for each entity pair (i, j) do
    Initialize tempTrust = 0;
    for each attribute D_k in D_{i,j} do
        tempTrust = tempTrust + w_k × f_k(C_{i,j});
    T_{i,j}^{opt} = tempTrust / n;
    Store T_{i,j}^{opt} in Ethereum mapping;
```

Temporal considerations were also integrated into our trust calculations, introducing a decay factor, δ, as shown in Equation (43).

$$T_{i,j}^{decay} = \delta \times T_{i,j}^{opt} + (1 - \delta) \times T_{i,j}^{prev} \tag{43}$$

Given the potential financial implications of deploying algorithms on Ethereum, a comprehensive gas analysis was imperative. We adopted the model presented in Equation (44) to quantify the gas expenditure for transactions.

$$G(x) = \alpha x + \beta \tag{44}$$

Furthermore, storage costs on Ethereum, represented by Equation (45), provided clarity on the economic feasibility of our system.

$$H(y) = \xi y + \zeta \tag{45}$$

External data acquisition remains a pivotal component of our system, necessitating interaction with decentralized oracles. Chainlink was adopted to facilitate this. The integration of trust computations with oracle feedback is represented by Equation (46).

$$T_{i,j}^{final} = T_{i,j}^{opt} + \gamma \cdot O_{i,j} \tag{46}$$

B. STORING TRUST VALUES ON ETHEREUM

The Ethereum blockchain, with its decentralized and immutable architecture, offers a robust foundation for recording computed trust values. By leveraging Ethereum's inherent characteristics, our system not only ensures the persistence of these values but also significantly bolsters the trust mechanism's credibility and reliability. Let us delve into the mathematical and algorithmic intricacies of this storage procedure.

Firstly, given an entity pair (i, j), the computed trust value $T_{i,j}$ is a scalar value representing the trust entity i places on entity j. Mathematically, for a set of entities E where $|E| = n$, the complete trust matrix T can be defined as:

$$T = \begin{bmatrix} T_{1,1} & T_{1,2} & \dots & T_{1,n} \\ T_{2,1} & T_{2,2} & \dots & T_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ T_{n,1} & T_{n,2} & \dots & T_{n,n} \end{bmatrix} \tag{47}$$

However, the Ethereum blockchain's storage mechanism is not amenable to direct matrix storage. Instead, a decentralized storage pattern using Ethereum's 'mapping' construct is more feasible. Specifically, the mapping connects pairs of addresses, which represent entities, to the corresponding trust value. The process of recording trust values post-computation into the Ethereum ledger is outlined in the Algorithm 5.

Algorithm 5 Storing Trust Values on Ethereum

Data: Entity pairs (i, j), Computed trust values $T_{i,j}$
Result: Stored trust values in Ethereum's decentralized ledger

foreach *entity pair* (i, j) **do**
    `trustStore[i][j]` $\leftarrow T_{i,j}$
    Validate storage integrity using a cryptographic hash
    Emit an event for the storage action

The cryptographic hash validation ensures that the stored value is identical to the computed value, offering an additional layer of data integrity. The emitted event aids external entities or systems in tracking changes and additions to the ledger.

C. GAS CONSUMPTION AND OPTIMIZATION

The Ethereum blockchain utilizes a mechanism termed as 'gas' to meter the computational effort required to execute operations, ranging from simple transfers to complex smart contract interactions. The pivotal role of gas ensures that participants in the ecosystem compensate for the computational energy required, and mitigates potential abuse scenarios such as infinite-loop attacks. During the implementation of our Blockchain-oriented Trust Management algorithm, careful consideration was dedicated to understanding and optimizing the gas consumption, as the iterative nature of our approach can lead to surges in operational costs.

To better grasp the cost implications, let us first define the function $G(x)$ that represents the gas consumption of a smart contract function, where x stands for the number of operations:

$$G(x) = \alpha x + \beta \tag{48}$$

In this expression, α denotes the gas consumed per operation, and β signifies the inherent overhead associated with initiating a smart contract function, irrespective of its complexity. To dissect our initial gas consumption, we analyze the summation involved in the calculation of trust values, as given by Equation 1. For an iterative summation over n entities, the associated gas cost G_{sum} can be modeled as:

$$G_{sum}(n) = \alpha_{sum} n + \beta_{sum} \tag{49}$$

where α_{sum} and β_{sum} represent the variable and fixed gas costs associated with the summation operation respectively. Recognizing the potential high gas costs from Equation 49, our strategy pivoted towards optimization. Minimizing storage operations, which are among the most gas-intensive operations in Ethereum, was our primary objective. We achieve this by restructuring the algorithm to use memory operations over storage wherever feasible, given that memory operations are transient and considerably cheaper. For a hypothetical scenario with m storage writes reduced to memory operations, the gas saving G_{save} can be represented as:

$$G_{save}(m) = \alpha_{store} m - \alpha_{memory} m \tag{50}$$

where α_{store} and α_{memory} represent the gas consumed per storage and memory operation respectively. Furthermore, by avoiding unnecessary computations and judiciously utilizing Ethereum's built-in functions which are gas-optimized, we introduced additional cost reductions. Specifically, we employed the 'keccak256' hashing function over traditional loops for certain repetitive operations, capitalizing on its fixed, lower gas cost. Given a set of p operations optimized in this manner, the total gas reduction G_{reduce} can be modeled as:

$$G_{reduce}(p) = \alpha_{loop} p - \alpha_{keccak} p \tag{51}$$

where α_{loop} and α_{keccak} denote the gas consumed per traditional loop operation and per 'keccak256' operation respectively. To visually elucidate the gas optimization process, consider the following algorithm:

D. INTEGRATION WITH DECENTRALIZED STORAGE: IPFS

A harmonious convergence of blockchain technology and decentralized storage mechanisms holds transformative potential, especially in an age where data integrity, accessibility, and decentralization are paramount. Within this context, Ethereum, though unparalleled in its capabilities for smart contracts and small data storage, confronts palpable limitations when tasked with preserving vast data volumes. These limitations are not just economic, in terms of escalating

Algorithm 6 Gas Optimization Procedure

Data: List of operations *Ops*, Predefined gas limits L
Result: Optimized list of operations *OptOps* with reduced gas consumption

```
OptOps ← ∅
foreach operation op in Ops do
    if G(op) > L then
        Optimize operation to use memory or 'keccak256'
    Append op to OptOps
return OptOps
```

gas costs, but also technical, as the blockchain would face undue bloat and scalability issues. The harmonized integration of metadata with IPFS and Ethereum has been methodically summarized in Algorithm 7. This procedure meticulously outlines the sequence of operations, from storing metadata on IPFS to updating the Ethereum contract with the corresponding hash references. By following such a systematic approach, we ensure consistency and verifiability in the integration process, reinforcing the robustness of our proposed solution.

Algorithm 7 Integration of Metadata With IPFS and Ethereum

Data: Metadata entries M, Ethereum contract instance E
Result: Hash references stored on Ethereum

```
foreach metadata m_i in M do
    h_i ← Store m_i on IPFS
    E.storeHash(h_i)
return "Metadata integrated successfully"
```

The InterPlanetary File System (IPFS), a decentralized storage protocol, offers a laudable solution to this quandary. Its design principle hinges on content-addressability, meaning data can be accessed based on its intrinsic content, not its location. This framework not only ensures resilience against data loss but also optimizes data retrieval in a distributed environment. To lay the foundation for our integration, it's imperative to first quantify the volume of metadata. Let us initiate our discourse by examining the representation and storage implications of individual metadata entries. Suppose M denotes a set encapsulating these metadata entries, represented as m_i, each corresponding to distinct virtual entities. The size occupied by each metadata entry, quantified in bytes, is expressed as $S(m_i)$. This is captured in Equation 52:

$$S(m_i) = \texttt{sizeof}(m_i) \tag{52}$$

Moving from individual entries to a collective perspective, it becomes imperative to evaluate the aggregated size of all metadata entries, denoted as S_{total}. This summation, spanning across all N entries, is depicted in Equation 53:

$$S_{total} = \sum_{i=1}^{N} S(m_i) \tag{53}$$

However, beyond the mere storage size, there arises the pivotal question of economic viability. Specifically, how costly is it to house such data directly on Ethereum? To address this, we introduce the function G_{store}, presented in Equation 54. This function is formulated to compute the gas requirement on Ethereum as a function of the data's size:

$$G_{store}(x) = \alpha x + \beta \tag{54}$$

By evaluating G_{store} against our cumulative metadata size, S_{total}, we glean insights into the prospective costs associated with direct Ethereum storage. This is elucidated in Equation 55:

$$G_{Ethereum} = G_{store}(S_{total}) \tag{55}$$

Contrastingly, the introduction of IPFS into our storage paradigm offers a nuanced strategy. Each metadata entry, when lodged within IPFS, yields a unique hash, denoted by $H(m_i)$, as delineated in Equation 56:

$$H(m_i) = \texttt{hash}(m_i) \tag{56}$$

Consequently, the cumulative storage footprint on Ethereum, now predominantly constituted of these hashes, shrinks considerably. This is captured in Equation 57:

$$S_{hash} = N \times \texttt{sizeof}(H(m_i)) \tag{57}$$

The resultant gas consumption, following this IPFS-centric approach, is elucidated in Equation 58:

$$G_{IPFS} = G_{store}(S_{hash}) \tag{58}$$

By juxtaposing the gas expenditures from Equations 55 and 58, the economic advantages of the IPFS-based model become unequivocally clear, as encapsulated in Equation 59:

$$G_{IPFS} \ll G_{Ethereum} \tag{59}$$

This comparative exposition underscores the heightened efficiency and cost-effectiveness of leveraging IPFS for metadata storage while relying on Ethereum primarily for its immutability and the storage of compact hashes.

E. EMPIRICAL ANALYSIS ON A REAL-TIME BLOCKCHAIN INFRASTRUCTURE

In our pursuit to substantiate the claims surrounding our system's efficiency and robustness, we embarked on rigorous experiments within an actual blockchain environment. This was paramount in gauging both the tangible performance attributes and affirming the system's resilience under genuine operational conditions. Our experimentation was orchestrated on the Ethereum Rinkeby testnet, ensuring an authentic blockchain experience devoid of the economic implications of the mainnet. We dedicated a node exclusively for this

TABLE 5. Summary of Experimental Results.

Metric	Average Value	Standard Deviation
Transaction Time	120 ms	15 ms
Gas Consumption	75,000 units	5,000 units
Data Retrieval Efficiency	90 ms	10 ms

TABLE 6. Results under high network congestion.

Metric	Average Value	Standard Deviation
Transaction Time	155 ms	20 ms
Gas Consumption	75,500 units	5,200 units
Data Retrieval Efficiency	95 ms	11 ms

purpose, enhancing result fidelity. The node was fortified with a quad-core processor, 16 GB RAM, and a solid-state drive to facilitate swift read-write operations, while network connectivity was maintained at a consistent bandwidth of 100 Mbps, eliminating potential bottlenecks.

In order to objectively assess the efficacy of our approach, it is paramount to delineate key performance indicators that resonate with the system's intended objectives. Accordingly, we defined the following metrics, which collectively encapsulate both the operational and economic facets of our blockchain-based solution:

- **Transaction Time:** This metric gauges the temporal efficiency of our system. Specifically, it measures the average duration required to process a transaction and secure it on the blockchain. A reduced transaction time would indicate an optimized smart contract and a more responsive system.
- **Gas Consumption:** As an economic barometer, this metric provides insight into the average gas costs associated with prevalent operations in our system. These operations encompass the computation of trust values, the archival of metadata on IPFS, and the extraction of hash references from Ethereum. An optimized gas consumption underscores the system's economic viability, particularly in a dynamic Ethereum gas price market.
- **Data Retrieval Efficiency:** Central to our integration with IPFS, this metric evaluates the expediency with which data can be retrieved from the decentralized storage and subsequently validated against its corresponding hash on Ethereum. A streamlined retrieval process, complemented by swift validation, exemplifies a harmonious interplay between the blockchain and IPFS in our approach.

F. RESULTS AND DISCUSSION

The empirical validation of our system stands crucial for underlining its applicability and efficiency in real-world scenarios. By benchmarking its performance on an actual blockchain platform, we could derive significant insights into its operability and potential areas for refinement. In the subsequent sections, we delve into a detailed analysis of the results, evaluated systematically across different scenarios. Table 5 provides an overarching perspective of the experimental results, covering three pivotal metrics: transaction time, gas consumption, and data retrieval efficiency.

1) SCENARIO ANALYSIS

The inherent complexities of real-world blockchain deployments necessitate a nuanced understanding of system performance across varied operational contexts. The scenario-based evaluation facilitates a deeper grasp of the system's resilience, adaptability, and efficiency under distinct circumstances, thereby offering a holistic perspective of its overall robustness. In the ensuing sections, we delineate the outcomes observed in multiple orchestrated scenarios, each designed to emulate potential real-world challenges and demands.

- **Scenario 1 (High Network Congestion):** Blockchain networks, like all distributed systems, are susceptible to variances in performance during periods of high traffic. One such scenario encountered during our experiments was that of heightened network congestion. This situation can arise due to a multitude of factors ranging from sudden surges in transactional demands to deliberate network spam attacks. Understanding system behavior under such conditions is paramount, given that performance bottlenecks during congestion can have cascading effects on user experience and system credibility. In our experimental setup, we simulated increased network activity to closely emulate conditions of high congestion. The results, as shown in Table 6, highlighted some intriguing findings:
  The observed escalation in the average transaction time underscores the network's throttled capacity to process transactions swiftly under duress. Meanwhile, the gas consumption also displayed a notable augmentation. This is potentially attributable to the network's dynamic fee mechanism, wherein heightened demand can inflate transactional costs. Lastly, the efficiency in data retrieval, although affected, showcased the resilience of the system, underscoring the fact that while transaction times may increase, the system's ability to access and validate data remains relatively robust. This scenario underscores the necessity for adaptive mechanisms within the protocol to account for such sporadic network challenges.
- **Scenario 2 (Voluminous Data Retrievals from IPFS):** Data retrieval performance, especially under conditions of sizeable data volumes, is a crucial aspect of any decentralized system. With our architecture's integration of IPFS for metadata storage, it was imperative to explore and analyze the system's behavior during bulk

TABLE 7. Results with voluminous data retrievals.

Metric	Average Value	Standard Deviation
Transaction Time	122 ms	17 ms
Gas Consumption	76,000 units	5,100 units
Data Retrieval Efficiency	105 ms	12 ms

TABLE 8. Results under frequent trust value computations.

Metric	Average Value	Standard Deviation
Transaction Time	125 ms	18 ms
Gas Consumption	80,000 units	5,300 units
Data Retrieval Efficiency	92 ms	11 ms

data retrieval scenarios. Such scenarios could arise during database migration, data backups, or mass validation processes. Our experiments simulated conditions where vast chunks of metadata were to be fetched in a short timeframe, thereby stress-testing the data retrieval pipeline. Table 7 summarizes the results for these voluminous data retrieval conditions:

Notably, transaction times remained comparatively consistent, reflecting the system's stable transaction processing abilities. The slight increment in gas consumption can be attributed to the fact that larger data retrievals often lead to larger transactional payloads and subsequently, higher computational demands on the Ethereum network. What stood out was the Data Retrieval Efficiency, which, at an average of 105 ms, only saw a modest increase despite the bulk data demands. This is indicative of IPFS's efficiency and scalability in handling large volumes of data. These findings illuminate the potential of IPFS when integrated with blockchain solutions, suggesting that our system is well-equipped to cater to use-cases where high-volume data retrievals are commonplace. Moreover, such resilience reinforces the choice of IPFS as a decentralized storage solution in blockchain-centric architectures.

- **Scenario 3 (Rapid Succession of Trust Value Computations):** Trust value computations, in any decentralized system, form the core of establishing credibility and trustworthiness of virtual entities. Particularly, in a system that bases its operation on a dynamic trust model, frequent calculations or recalculations of trust values can emerge due to a myriad of reasons: increased interactions among entities, sporadic behavior of some entities leading to trust reassessment, or periodic updates enforced by the system. As a result, assessing the system's performance under intensive trust computation scenarios is imperative.

  In our experimentation, we simulated an environment characterized by rapid successive trust computations. This involved frequent updating of trust scores based on simulated interactions, feedback, and behavioral assessments, thereby pushing the boundaries of the trust evaluation algorithm. The results, as enumerated in Table 8, provide a perspective on the system's efficiency under such intensive conditions:

  The Transaction Time, being consistent with previous scenarios, indicates that the trust computation mechanism does not introduce significant latency to transaction processing. A noteworthy observation is the elevated gas consumption. This can be inferred as a direct result of the computational overhead introduced by repetitive trust value recalculations. Nevertheless, the gas consumption remains within acceptable limits, highlighting the efficiency of the trust computation algorithm. The data retrieval efficiency was also commendable, showcasing the system's ability to access required metadata promptly even under computational duress. In essence, the results cement the system's capability to handle trust computations proficiently, even when they are demanded in rapid succession. This resilience underscores the robustness of the proposed trust evaluation algorithm and its suitability for real-world deployment in dynamic decentralized ecosystems.
- **Scenario 4 (Mixed Operational Load):** The hallmark of an efficient blockchain-based system lies in its ability to effectively manage heterogeneous transactional demands without compromising on performance. Real-world deployments often entail a diverse set of operations being executed concurrently, ranging from simple data storage and retrievals, trust computations, to intricate smart contract invocations. It is in such mixed operational environments that the true robustness and efficiency of a system are tested. In this scenario, we simulated an environment replete with a blend of transaction types. This involved:
  - High-frequency data storage and retrieval operations from IPFS.
  - Randomized trust value computations, mimicking sporadic interactions among entities.
  - Execution of diverse smart contract functions, including but not limited to, metadata storage, hash retrievals, and complex data manipulations.

  The outcomes of this rigorous testing regimen are encapsulated in Table 9:

  Observations from this scenario are multifaceted:

  1) The Transaction Time, albeit slightly elevated, remains within acceptable bounds. This illustrates the system's optimized transaction processing pipeline that can handle diverse operations simultaneously.
  2) Gas consumption, while commensurate with the other scenarios, reveals the system's capacity to

TABLE 9. Results under mixed operational load.

Metric	Average Value	Standard Deviation
Transaction Time	130 ms	19 ms
Gas Consumption	77,500 units	5,250 units
Data Retrieval Efficiency	93 ms	10 ms

TABLE 10. Results under adversarial conditions.

Metric	Average Value	Standard Deviation
Transaction Time	140 ms	21 ms
Gas Consumption	78,000 units	5,400 units
Data Retrieval Efficiency	100 ms	13 ms

maintain computational efficiency, even with a slew of distinct operations.

3) The Data Retrieval Efficiency metric stands testament to the system's consistent ability to fetch and validate data, regardless of the operational backdrop.

- **Scenario 5 (External Interference and Attacks):** The resilience of a blockchain-based system, particularly in the domain of decentralized applications, is gauged not just by its ability to process transactions efficiently, but also by its fortitude against external threats and adversarial conditions. A gamut of potential attacks, ranging from Sybil and DDoS attacks to more sophisticated threats like eclipse attacks, can target the integrity, availability, and consistency of a blockchain system.
  In this scenario, our intent was to recreate a hostile operational environment where the system would be subjected to various external interferences, both in terms of network disruption and attempts to compromise the integrity of the data. Specific adversarial conditions simulated included:
  - Intermittent network disruptions, mimicking DDoS attacks.
  - Injection of malicious nodes in the network, attempting to introduce false transactions and corrupt the ledger.
  - Attempts to alter stored metadata in IPFS.
  - Targeted attacks on smart contracts, exploiting known vulnerabilities.

  Table 10 succinctly captures the performance metrics under these stringent conditions:
  The derived observations from this scenario shed light on multiple fronts:
  1) The slightly augmented Transaction Time is a testament to the added computational overhead of defending against external threats. Despite the adversarial conditions, the increase remains reasonably contained.
  2) The Gas Consumption metric, while elevated, is indicative of the system's strategic resource allocation to both maintain operational integrity and thwart malicious endeavors.
  3) Data Retrieval Efficiency showcases the robust design of the integration between Ethereum and IPFS, where even under threat, data integrity and retrieval mechanisms remain largely unhampered.

The results, summarized in tables, indicate that our implementation is both efficient and economical. Transaction times were consistently low, demonstrating the system's responsiveness. Gas consumption, a crucial determinant of operational feasibility, remained within acceptable limits, underscoring the efficacy of our optimization strategies. The data retrieval efficiency further highlights the seamless integration between Ethereum and IPFS in our architecture.

VII. DISCUSSION

Over the past few years, there has been an unprecedented surge in the evolution of metaverse technology. With this rapid growth, a pressing challenge arises: how to ensure the safe and reliable interaction of virtual entities within the metaverse. It is here that the environment of trusted blockchain construction finds its relevance and is poised to make a significant contribution. In the context of the metaverse, blockchain technology has the potential to provide robust, secure, and trustworthy systems that can enable and facilitate seamless interactions between virtual entities. Decentralization, the essence of blockchain, empowers each entity with the ability to control its data, transactions, and interactions, ensuring a level of fairness and trust that is hard to achieve in centralized systems.

Our research has shown that the proposed DSCM, a decentralized dispute resolution mechanism can significantly enhance the performance of metaverse systems. Its accuracy in detecting malicious nodes, the rapidity of dispute resolution, and its resilience against various forms of attacks bear testimony to this. Selecting the most appropriate simulation parameters was a crucial step to ensure the accuracy of our experimental findings. The values of these parameters were chosen based on conventional practices, relevant research literature, and the constraints of our experimental setup. In addition to adhering to established standards of current blockchain networks, we carried out a comprehensive series of preliminary tests for parameters unique to our proposed model. These preliminary tests were aimed at identifying the parameter values that optimize our model's efficiency. In terms of detection rate and dispute resolution accuracy, our simulations reveal that the proposed DSCM significantly outperforms the existing approaches. The detection rate of DSCM was found to be 95%, which is significantly higher than the 80% and 75% of BTCGS and MSBC-CTrust. Time efficiency, another key performance metric, also reflects favorably on our proposed mechanism. The average dispute resolution time for DSCM is calculated

to be 25 seconds, compared to the 40 seconds and 55 seconds of BTC-Trust and MSBC-CTrust. However, aligning the pace of trusted blockchain construction with the rapid metaverse development presents its own set of challenges. As the metaverse continues to evolve, we expect to see increasingly sophisticated forms of attacks, which call for more advanced and dynamic dispute resolution mechanisms. Hence, the trust models and dispute resolution mechanisms we develop for the metaverse need to be robust and adaptable.

In addition, the metaverse is expected to host an incredibly diverse range of interactions. From casual social interactions to high-stakes business transactions, the demands on a trust system will vary considerably. Balancing the trade-offs between security, speed, and resource consumption will be a key challenge. Furthermore, as the metaverse expands, the size and complexity of the blockchain networks underpinning it will also increase. This expansion will inevitably put a strain on the scalability of the blockchain systems. Techniques to ensure blockchain scalability, such as sharding and off-chain transactions, need to be incorporated into the design of the trust systems. Keeping pace with the rapid development of metaverse technology requires continuous innovation in the construction of trusted blockchains. This innovation encompasses improving the accuracy and efficiency of dispute resolution mechanisms, making the trust systems adaptable to the evolving threat landscape, and addressing the scalability concerns associated with large-scale blockchain networks.

VIII. CONCLUSION

In this research, we advanced a robust paradigm for trust management in decentralized networks, amalgamating trust scores with reputation scores to adeptly pinpoint and counteract an array of attacks including Sybil, swearing, and on-off attacks. This was achieved through the cultivation of a sophisticated reputation system that computes a node's reputation score derived from feedback amassed from peer nodes. Complementing this, we instantiated a trust model that determines the trust score of each node, grounded on its reputation score and the credibility of its participated transactions. Implementation outcomes highlighted a marked superiority of our methodology over prevalent systems such as BTC-Trust and MSBC-CTrust. Notably, our system exhibited an accuracy of 98% in intercepting on-off attacks, 94% for good-kissing attacks, 97% in identifying bad-mouth attacks, and 95% in detecting Sybil attacks. These commendable accuracies were accompanied by expedient mitigation times: averaging 30 seconds for on-off attacks, 40 seconds for good-kissing attacks, 28 seconds for bad-mouth attacks, and 60 seconds for Sybil attacks. Our research also integrated the intricacies of blockchain implementation, underscoring the economic ramifications of direct storage on Ethereum versus an IPFS-assisted approach. Through this exploration, the dual virtues of data accessibility and cost efficiency became evident.

REFERENCES

[1] R. Chengoden, N. Victor, T. Huynh-The, G. Yenduri, R. H. Jhaveri, M. Alazab, S. Bhattacharya, P. Hegde, P. K. R. Maddikunta, and T. R. Gadekallu, "Metaverse for healthcare: A survey on potential applications, challenges and future directions," *IEEE Access*, vol. 11, pp. 12765–12795, 2023.

[2] A. Audrezet and B. Koles, "Virtual influencer as a brand avatar in interactive marketing," in *The Palgrave Handbook of Interactive Marketing*. Cham, Switzerland: Springer, 2023, pp. 353–376.

[3] B. M. Demaerschalk, J. E. Hollander, E. Krupinski, J. Scott, D. Albert, Z. Bobokalonova, M. Bolster, A. Chan, L. Christopherson, J. D. Coffey, S. Edgman-Levitan, J. Goldwater, E. Hayden, C. Peoples, K. L. Rising, and L. H. Schwamm, "Quality frameworks for virtual care: Expert panel recommendations," *Mayo Clinic Proc., Innov., Quality Outcomes*, vol. 7, no. 1, pp. 31–44, Feb. 2023.

[4] R. Das and M. Dwivedi, "Cluster head selection and malicious node detection using large-scale energy-aware trust optimization algorithm for HWSN," *J. Reliable Intell. Environ.*, early access, 2023, doi: 10.1007/s40860-022-00200-6.

[5] K. Y. Yap, H. H. Chin, and J. J. Klemeš, "Blockchain technology for distributed generation: A review of current development, challenges and future prospect," *Renew. Sustain. Energy Rev.*, vol. 175, Apr. 2023, Art. no. 113170.

[6] L. Gauder, L. Pepino, P. Riera, S. Brussino, J. Vidal, A. Gravano, and L. Ferrer, "Towards detecting the level of trust in the skills of a virtual assistant from the user's speech," *Comput. Speech Lang.*, vol. 80, May 2023, Art. no. 101487.

[7] C. Lewis, N. Li, and V. Varadharajan, "Targeted context based attacks on trust management systems in IoT," *IEEE Internet Things J.*, vol. 10, no. 14, pp. 12186–12203, Jul. 2023.

[8] A. Alamsyah, S. Widiyanesti, P. Wulansari, E. Nurhazizah, A. S. Dewi, D. Rahadian, D. P. Ramadhani, M. N. Hakim, and P. Tyasamesi, "Blockchain traceability model in the coffee industry," *J. Open Innov. Technol., Market, Complex.*, vol. 9, no. 1, Mar. 2023, Art. no. 100008.

[9] L. Zhang, M. A. Anjum, and Y. Wang, "The impact of trust-building mechanisms on purchase intention towards metaverse shopping: The moderating role of age," *Int. J. Hum.-Comput. Interact.*, 2023, doi: 10.1080/10447318.2023.2184594.

[10] S. Ali, Abdullah, T. P. T. Armand, A. Athar, A. Hussain, M. Ali, M. Yaseen, M.-I. Joo, and H.-C. Kim, "Metaverse in healthcare integrated with explainable AI and blockchain: Enabling immersiveness, ensuring trust, and providing patient data security," *Sensors*, vol. 23, no. 2, p. 565, Jan. 2023.

[11] A. Sathya, "Blockchain: The foundation of trust in Metaverse," in *Recent Advances in Blockchain Technology: Real-World Applications*. Cham, Switzerland: Springer, 2023, pp. 117–129.

[12] Z. Lin, P. Xiangli, Z. Li, F. Liang, and A. Li, "Towards Metaverse manufacturing: A blockchain-based trusted collaborative governance system," in *Proc. 4th Int. Conf. Blockchain Technol.*, Mar. 2022, pp. 171–177.

[13] S. Badruddoja, R. Dantu, Y. He, M. Thompson, A. Salau, and K. Upadhyay, "Trusted AI with blockchain to empower Metaverse," in *Proc. 4th Int. Conf. Blockchain Comput. Appl. (BCCA)*, Sep. 2022, pp. 237–244.

[14] K. Gai, S. Wang, H. Zhao, Y. She, Z. Zhang, and L. Zhu, "Blockchain-based multisignature lock for UAC in Metaverse," *IEEE Trans. Computat. Social Syst.*, vol. 10, no. 5, pp. 2201–2213, Oct. 2023.

[15] 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.

[16] X. Wu, S. Wei, Z. Zhang, and P. Lv, "Master-slave blockchain framework: Cross-domain trust management mechanism using trust ticket," *IEEE Trans. Netw. Service Manag.*, vol. 20, no. 3, pp. 2830–2844, Sep. 2023.

[17] M. T., K. Makkithaya, and V. G. Narendra, "A trusted IoT data sharing and secure oracle based access for agricultural production risk management," *Comput. Electron. Agricult.*, vol. 204, Jan. 2023, Art. no. 107544.

[18] M. Xu, Y. Guo, Q. Hu, Z. Xiong, D. Yu, and X. Cheng, "A trustless architecture of blockchain-enabled Metaverse," *High-Confidence Comput.*, vol. 3, no. 1, Mar. 2023, Art. no. 100088.

[19] H. Moudoud and S. Cherkaoui, "Multi-tasking federated learning meets blockchain to foster trust and security in the Metaverse," *Ad Hoc Netw.*, vol. 150, Nov. 2023, Art. no. 103264.

[20] A. Corne, V. Massot, and S. Merasli, "The determinants of the adoption of blockchain technology in the tourism sector and Metaverse perspectives," *Inf. Technol. Tourism*, vol. 25, no. 4, pp. 605–633, Dec. 2023.
[21] D. Mourtzis, J. Angelopoulos, and N. Panopoulos, "Blockchain integration in the era of industrial Metaverse," *Appl. Sci.*, vol. 13, no. 3, p. 1353, Jan. 2023.
[22] S. Islam, "Trust in digital asset transactions in a web 3 based metaverse," Oulu Bus. School, Univ. Oulu, Oulu, Finland, May 2023.
[23] A. S. Rajawat, S. B. Goyal, A. Goyal, K. Rajawat, M. S. Raboaca, C. Verma, and T. C. Mihaltan, "Enhancing security and scalability of Metaverse with blockchain-based consensus mechanisms," in *Proc. 15th Int. Conf. Electron., Comput. Artif. Intell. (ECAI)*, Jun. 2023, pp. 1–6.
[24] Y. Ren, Z. Lv, N. N. Xiong, and J. Wang, "HCNCT: A cross-chain interaction scheme for the blockchain-based metaverse," *ACM Trans. Multimedia Comput., Commun., Appl.*, early access, Apr. 2023, doi: 10.1145/3594542.
[25] C. Wang, Z. Cai, D. Seo, and Y. Li, "TMETA: Trust management for the cold start of IoT services with digital-twin-aided blockchain," *IEEE Internet Things J.*, early access, 2023, doi: 10.1109/JIOT.2023.3285108.

KAMRAN AHMAD AWAN received the B.S. and M.S. degrees in computer science from the Department of Information Technology, The University of Haripur, Pakistan, in 2015 and 2019, respectively, where he is currently pursuing the Ph.D. degree in computer science. His research interests include the trust management in Internet of Things, blockchain, security in metaverse, and information security.

IKRAM UD DIN (Senior Member, IEEE) received the Ph.D. degree in computer science from the School of Computing, Universiti Utara Malaysia (UUM), in 2016. Currently, he is an Associate Professor with the Department of Information Technology, The University of Haripur. He has 13 years of teaching and research experience in different universities/organizations. His current research interests include traffic measurement and analysis for monitoring quality of service, mobility, cache management in information-centric networking, and the Internet of Things. He also served as the IEEE UUM Student Branch Professional Chair.

AHMAD ALMOGREN (Senior Member, IEEE) received the Ph.D. degree in computer science from Southern Methodist University, Dallas, TX, USA, in 2002. He is currently a Professor with the Computer Science Department, College of Computer and Information Sciences (CCIS), King Saud University (KSU), Riyadh, Saudi Arabia. He is also the Director of the Cyber Security Chair, CCIS, KSU. Previously, he was the Vice Dean of the Development and Quality with CCIS. He was also the Dean of the College of Computer and Information Sciences and the Head of the Academic Accreditation Council with Al-Yamamah University. His research interests include mobile-pervasive computing and cyber security. He served as the General Chair for the IEEE Smart World Symposium and a Technical Program Committee Member in numerous international conferences/workshops, such as IEEE CCNC, ACM BodyNets, and IEEE HPCC.

BYUNG SEO-KIM (Senior Member, IEEE) received the B.S. degree in electrical engineering from Inha University, Incheon, South Korea, in 1998, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Florida, in 2001 and 2004, respectively. His Ph.D. study was supervised by Dr. Yuguang Fang. Between 1997 and 1999, he was with Motorola Korea Ltd., Paju, South Korea, as a Computer Integrated Manufacturing (CIM) Engineer in advanced technology research and development (ATR&D). From January 2005 to August 2007, he was with Motorola Inc., Schaumburg, IL, USA, as a Senior Software Engineer in networks and enterprises. His research focuses with Motorola Inc., were designing protocol and network architecture of wireless broadband mission critical communications. From 2012 to 2014, he was the Chairperson of the Department of Software and Communications Engineering, Hongik University, South Korea, where he is currently a Professor. His research interests include the design and development of efficient wireless/wired networks, including link-adaptable/cross-layer-based protocols, multi-protocol structures, wireless CCNs/NDNs, mobile edge computing, physical layer design for broadband PLC, and resource allocation algorithms for wireless networks. He served as the General Chair for Third IWWCN 2017 and a TPC Member for the IEEE VTC 2014-Spring and the EAI FUTURE2016, and ICGHIT 2016 2019 conferences. He served as a Guest Editor for special issues for *International Journal of Distributed Sensor Networks* (SAGE), IEEE Access, *Sensors* (MDPI), and *Journal of the Institute of Electrics and Information Engineers*. He was also served as a member for Sejong-City Construction Review Committee and Ansan-City Design Advisory Board. His work has appeared in around 174 publications and 25 patents. He is also an Associative Editor of IEEE Access.