electronics

MDPI

Article

Enhancing Performance and Security in the Metaverse: Latency Reduction Using Trust and Reputation Management

Kamran Ahmad Awan [1], Ikram Ud Din [1,*], Ahmad Almogren [2] and Byung-Seo Kim [3,*]

[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 30016, Republic of Korea
* Correspondence: ikramuddin205@yahoo.com (I.U.D.); jsnbs@hongik.ac.kr (B.-S.K.)

Abstract: In the rapidly evolving landscape of distributed systems, security stands as a significant challenge, especially in the face of network node attacks. Such threats introduce profound complexities into the dynamics of security protocols, trust management, and resource allocation, issues further amplified by the metaverse's exponential growth. This paper proposes an innovative solution, offering unique technical contributions to address these multi-faceted challenges. We unveil a trust-based resource allocation framework designed to facilitate the secure and efficient sharing of computational resources within the metaverse. This system has the potential to markedly diminish latency, thereby enhancing overall performance. In parallel, we introduce a reputation system that systematically monitors latency across a spectrum of metaverse entities, providing valuable insights for making informed resource allocation decisions. Moreover, we advocate for a decentralized trust management system, specifically designed to withstand potential security breaches without reliance on a centralized authority. This significantly fortifies both system security and user trust. Alongside this, we unveil an inventive proof-of-trust consensus mechanism that fosters trust and collaboration among metaverse entities during resource allocation, thereby cultivating a more secure ecosystem. Our proposed model poses a robust challenge to malicious entities, and it substantially bolsters the security architecture. The simulation results lend substantial credence to the effectiveness of our approach, demonstrating significant improvements in latency reduction, scalability, and the detection of malicious nodes, thereby outperforming existing methodologies.

Keywords: trust management; latency reduction; metaverse; throughput; consensus mechanism; reputation management; trustworthiness; privacy preservation

Citation: Awan, K.A.; Ud Din, I.; Almogren, A.; Kim, B.-S. Enhancing Performance and Security in the Metaverse: Latency Reduction Using Trust and Reputation Management. *Electronics* **2023**, *12*, 3362. https://doi.org/10.3390/electronics12153362

Academic Editor: Martin Reisslein

Received: 22 May 2023
Revised: 29 July 2023
Accepted: 3 August 2023
Published: 6 August 2023

1. Introduction

The conception of the metaverse [1], which postulates a virtual environment that enables interaction between physical and digital constituents [2], has taken tangible form, owing to the proliferation of virtual reality (VR) and augmented reality (AR) technologies [3]. This transcendent paradigm shift in human–digital interaction has piqued the interest of technologists and futurists alike [4], extending its tendrils into a broad array of online environments. Today, the metaverse's realm encompasses social networks [5], gaming [6], virtual real estate [7], digital art exhibitions [8], and more, attracting substantial attention and investment from major technological corporations and startups [9].

As a potential successor to smartphones and personal computers, the metaverse's vast influence touches upon numerous sectors, including communication [10], education [11], entertainment [12], and commerce [13]. Nevertheless, with this evolutionary stride towards a digitally intertwined reality comes a plethora of challenges. Among these, latency [14], defined as the delay between a user's action and the system's response [15], is of paramount concern. High latency can drastically compromise user experience [16], underscoring the

urgent need for efficient latency mitigation strategies to enhance the metaverse's overall performance [17].

Prior endeavors to alleviate latency have largely focused on the enhancement of network infrastructure and optimization of content delivery [18], processes that often necessitate significant resources yet yield marginal returns. This paper ventures into an unexplored territory: the incorporation of distributed trust management strategies. Unlike its centralized counterpart, which frequently poses as a bottleneck and a vulnerability hotspot, distributed trust management decentralizes trust decisions across the network, potentially ameliorating system efficiency and reducing latency. Our study introduces an avant garde trust-based strategy, meticulously designed to address latency in the metaverse and bolster performance. The contributions to the metaverse performance optimization in our study include the following:

- Trust-based resource allocation: A novel methodology facilitating the secure and efficient sharing of computational resources among various metaverse entities.
- Reputation-based latency reduction: A comprehensive rating system that routinely monitors latency across distinct metaverse entities, aiming to minimize user-experienced latency.
- Decentralized trust management: A resilient system, devoid of centralized authority, specifically designed to resist attacks and reinforce user trust.
- Proof-of-trust consensus mechanism: An inventive mechanism enabling entities within the metaverse to establish trust and coordinate resource allocation.

The remainder of the paper is structured as follows: Section 2 elaborates on the context of trust management in the metaverse, dissecting the limitations of contemporary methods. In Section 3, we explicate our proposed technique dedicated to enhancing metaverse performance through trust management strategies for latency reduction. Section 4 provides a comprehensive analysis of our method's efficacy, encompassing a comparative evaluation with extant methods. Finally, in Section 5, we explore prospective research trajectories and summarize our findings, shedding light on their implications for the future evolution of the metaverse.

2. Related Work

In the burgeoning field of metaverse-based Internet of Things (IoT), rapid advancement has resulted in an unprecedented proliferation of interconnected devices. This expansion, while promising, has led to certain challenges, primarily latency induced by network congestion or capacity constraints, resulting in possible data inaccuracies or loss. To mitigate these concerns, researchers have explored potential solutions, such as mobile edge computing, service deployment strategies, low-latency wireless communication, hybrid edge–cloud systems, and channel-sharing strategies. This section presents an extensive review of these diverse latency reduction methodologies and their comparative assessment as detailed in Table 1.

Our study explores an innovative methodology to address latency in the metaverse through the integration of trust management strategies and a reputation system, which distinguishes our approach from previous work in several ways. Prior work [19–22] predominantly focused on conventional techniques for latency reduction, such as service placement algorithms, improvements in wireless communication technologies, and edge computing. While these methods have made substantial contributions to minimizing latency, they often do not account for the unique characteristics of the metaverse, particularly the significant reliance on trust due to the absence of a centralized authority.

Table 1. A comparative analysis of the existing approaches.

Ref.	Methods	Key Findings	Focus
[19]	Latency Reduction	Implemented a service placement algorithm to reduce service response time and enhance IoT performance	Reduced service response time
[20]	Latency Reduction	Systematic literature review exploring techniques for reducing latency in IoT and cloud computing for real-time data transmission	Improvement techniques for latency reduction
[21]	Latency and Reliability	Detailed study of technologies enabling low-latency and reliable communication in IoT	Enabling technologies for low-latency and reliable communication
[22]	Latency Reduction	Mobile edge computing proposed to reduce latency in green IoT	Reduced latency for green IoT
[23]	Latency Reduction	Introduced a channel-sharing approach with artificial jamming to reduce latency in secure wireless federated learning	Latency-oriented secure wireless federated learning
[24]	Latency Reduction	Explored narrowband IoT to enhance vehicular communication performance	Enhanced vehicular communication performance
[25]	Latency and Energy Consumption	Edge computing used to understand latency and energy consumption of convolutional neural network (CNN) models from an IoT edge perspective	Latency and energy consumption of CNN models
[26]	Latency Reduction	Adaptive network access proposed for industrial IoT based on statistical learning to reduce latency	Adaptive network access for industrial IoT
[27]	Latency Reduction	Introduced channel-aware latency tail taming to improve latency in industrial IoT	Improved latency in industrial IoT
[28]	Networking Optimization	Proposed a hybrid edge–cloud system for optimizing networking service components using IoT	Networking service components optimization using IoT
[29]	Latency and Security	Developed a new low-latency lightweight block cipher for information security and cryptography	New low-latency lightweight block cipher

Our work diverges from these traditional approaches by incorporating trust management to optimize resource allocation. This is achieved through the novel use of a reputation-based system that not only facilitates effective sharing of computational resources but also ensures secure interactions among various entities within the metaverse. To our knowledge, this is the first study to exploit such a trust-based approach for latency reduction in the metaverse. Moreover, we also introduce a decentralized trust management system, independent of a centralized authority, which adds resilience against potential security threats. This feature further accentuates the uniqueness of our contribution, as most previous studies [23–25] focused on latency reduction in IoT or edge networks, often reliant on centralized control mechanisms.

A service placement system for the Internet of Things was presented by Velazquez et al. [19] that takes service–device relationships into account in order to reduce latency. The plan's goal is to reduce network latency by relocating the service physically closer to the end user's gadget. On the other hand, Shukla et al. [20] conducted a systematic literature review on improving latency for real-time data transfer in IoT and cloud computing. They identified key factors affecting latency in these systems, including device limitations, network topology, and protocol choice.

In [30], a new distributed messaging system called 'Mez', was designed to process delay-sensitive multi-camera vision at the edge of IoT networks. The proposed approach demonstrates the need for a low-latency messaging system that can process large amounts of data generated by multiple cameras in real time. This article contains a detailed description of the system architecture, experimental results, and a comparative analysis with other messaging systems. Results show that Mez outperforms existing messaging systems in terms of end-to-end latency and message delivery speed. Mez has the potential to become

a major disruptive technology with applications in large-scale machine vision, real-time systems and virtual reality.

Horse et al. [21] researched fundamental and enabling technologies for high-reliability, low-latency wireless communication for IoT. They presented a detailed analysis of the challenges and solutions for reliable low-latency communications, including channel allocation, power control, and modulation. In addition, Zhang et al. [22] proposed mobile edge computing and network technologies that enable low-latency communication and green computing in IoT. This approach allows edge devices to perform computing tasks traditionally performed in the cloud, significantly reducing latency and power consumption in IoT networks.

Recently, Wang et al. [23] proposed a latency-oriented approach for secure wireless federated learning that applies artificial jamming to optimize channel sharing in his IoT network. They aimed to address the challenges of maintaining privacy and security in federated learning, which is becoming increasingly important in IoT. Further, Hamarsheh et al. [24] introduced a narrowband IoT system that improves vehicle communication performance with low latency and high reliability. They used narrowband technology to overcome the limitations of traditional broadband technology in vehicular communications.

Hauschild and Hellbruck [25] examined the latency and power consumption of convolutional neural network models from the IoT edge perspective. They aimed to minimize the latency and power consumption of these models for edge devices by investigating various optimization techniques. Raza et al. [26] proposed a statistical learning-based adaptive network access approach for industrial IoT that enables efficient data transmission in demanding environments. We used machine learning techniques to optimize network access and ensure reliable, low-latency communication.

In [27], a channel-aware latency tail-taming approach that effectively reduces end-to-end latency in industrial IoT was explored. They used a new model to identify the causes of latency tails in IoT networks and optimize channel allocation to reduce latency. Fundal et al. [31] explored barriers to the adoption of IoT-based precision agriculture practices. They found that farmers' concerns about privacy, lack of interoperability, and inadequate training were the main challenges for IoT adoption in precision agriculture. Finally, Pal et al. [28] announced a hybrid edge–cloud system for optimizing network service components with IoT. They proposed a new architecture that integrates edge and cloud computing to optimize IoT network performance by reducing network latency and power consumption.

3. Proposed Framework

The framework we propose for addressing latency in the metaverse consists of four main interdependent components. Each of these components is intricately designed to collectively enhance metaverse performance while simultaneously minimizing latency. In the following subsections, we elucidate the functionality and interplay of each component within the proposed framework.

3.1. Framework Design

The framework's design integrates four main components in a synergistic manner: trust-centric resource allocation, latency reduction via reputation mechanism, decentralized trust management, and trust verification via consensus mechanism. These components form an all-inclusive blueprint for latency reduction in the metaverse ecosystem. The proposed framework operates on a decentralized network model with various entities in the metaverse—which could be users, servers, or other nodes—interacting and exchanging resources. These entities utilize trust-centric resource allocation and latency reduction via reputation mechanism to ensure seamless operations.

- Trust-centric resource allocation: Trust-centric resource allocation serves as the initial stage in the proposed framework. Each entity in the network is assigned a trust score, which is computed based on previous interactions and behavioral patterns. Resource allocation is then guided by these trust scores, with entities boasting higher trust scores receiving priority in resource allocation. This strategic prioritization guarantees that reliable entities are well equipped with the necessary resources, thus potentially reducing latency and bolstering the overall performance of the metaverse.
- Latency reduction via reputation mechanism: Operating in parallel with the trust-centric resource allocation, the framework also employs latency reduction via reputation mechanism. The latency of each entity is persistently monitored, and the collected data are utilized to modify a latency reputation score. This score significantly influences resource allocation, with entities exhibiting lower latency scores receiving preference, thereby aiding in the reduction in the overall network latency.
- Decentralized trust management: The system manages trust scores and latency reputation scores in a decentralized manner. Each node participates in the upkeep and modification of these scores, eliminating the requirement for a centralized authority. This decentralization not only bolsters the system's resilience against potential attacks but also fosters user trust in the absence of a centralized reliable entity.
- Trust verification via consensus mechanism: The trust verification via consensus mechanism constitutes the final component of the framework. It is employed to modify trust scores and latency reputation scores. When a transaction is conducted between entities, the entities involved share the details of the transaction with the network. The network nodes then utilize the trust verification via consensus mechanism to reach a consensus on the transaction's outcome, thereby updating the trust scores and latency reputation scores correspondingly.

The proposed framework offers a groundbreaking solution for latency reduction within the metaverse. By integrating trust-centric resource allocation, latency reduction via reputation mechanism, decentralized trust management, and trust verification via consensus mechanism, it optimizes resource allocation, minimizes latency, and ultimately enhances the user experience within the metaverse.

3.2. Trust-Centric Resource Allocation

The trust-centric resource allocation (TCRA) protocol operates on a resilient trust management scheme, an indispensable facet enabling the secure and efficient allocation of computational resources across a broad spectrum of entities within the metaverse. The core principle of this system lies in the quantification of trust, represented as a composite score. This score emanates from a comprehensive evaluation of various parameters associated with each entity, namely, honesty, reputation, cooperative behavior, and demonstrated competence as illustrated in Algorithm 1.

The TCRA algorithm commences by initializing a set of n entities within the metaverse. These entities could manifest as users, server nodes, or any other interactive elements within the digital realm. Concurrently, a total of m interactions among these entities is also initialized, encompassing the various communications and transactions that may transpire within the metaverse. In terms of trust calculation, a multi-factor approach is employed, considering honesty (H_{ij}), reputation (R_{ij}), cooperativeness (C_{ij}), and competence (K_{ij}) as constituent elements of the trust score (T_{ij}). Each of these factors is evaluated for every pair of entities (i, j) within the system. The trust score is then calculated using the established weights w_h, w_r, w_c, and w_k for honesty, reputation, cooperativeness, and competence, respectively, as shown in Equation (1). Following this, the algorithm determines the fairness coefficient F_i for each entity i, which is an essential element for ensuring equitable resource distribution. This coefficient, derived using Equation (6), serves as a balancing factor in resource allocation, factoring in the disparity among different entities' capabilities and needs.

Algorithm 1: Trust-centric resource allocation algorithm.

Result: Trust-centric resource allocation (TCRA)
Initialize n entities, m total interactions, w_h, w_r, w_c, w_k, β, R_{total}, α, γ;
for *each entity* i **do**
  for *each entity* j **do**
    Compute honesty H_{ij}; Evaluate reputation R_{ij}; Determine cooperativeness C_{ij}; Assess competence K_{ij}; Derive trust score T_{ij} using Equation (1);
  end
  Determine fairness coefficient F_i using Equation (6); Allocate resources R_i using Equation (5); Measure latency L_i using Equation (7); **for** *each interaction* **do**
    Monitor changes in T_i, denoted as ΔT_i; Adjust R_i proportionally using Equation (12);
  end
end

Next, the algorithm allocates resources R_i to each entity based on the fairness coefficient and the overall trust score. This allocation process, guided by Equation (5), is designed to provide more resources to entities with higher trust scores, thereby promoting trustworthiness and reliability in the system. Further, the algorithm estimates the latency L_i for each entity using Equation (7). The latency measurement here serves as an integral part of the latency reduction approach of the system. Lower latency ensures smoother, quicker interactions in the metaverse, thus enhancing user experience and system efficiency. Finally, during each interaction within the metaverse, the algorithm monitors changes in trust scores denoted by ΔT_i. If a significant change in the trust score occurs, the algorithm adjusts the entity's resource allocation proportionally using Equation (12). This adaptive feature allows the TCRA protocol to respond dynamically to the evolving behavior of entities, maintaining optimal resource distribution, latency reduction, and overall system performance.

This protocol emphasizes the critical role of trust in managing the allocation of computational resources within a heterogeneous metaverse environment. By assessing individual behavioral characteristics and translating these into a trust score, it ensures an equitable distribution of resources. This process serves to improve the overall performance and reduce latency within the metaverse, while simultaneously preserving the system's security. Moreover, by dynamically adjusting the resource allocation based on the changes in trust scores, the TCRA protocol maintains an adaptable environment that mirrors the evolving dynamics of interactions among the entities.

3.2.1. Constructing Trust Metric

We introduce a trust metric T, quantifying the reliability of an entity. This metric constitutes a composite function encompassing four essential behavioral aspects: honesty H, reputation R, cooperativeness C, and competence K. The trust score T_{ij}, reflecting the trustworthiness of entity i as judged by entity j, is mathematically encapsulated as

$$T_{ij} = w_h H_{ij} + w_r R_{ij} + w_c C_{ij} + w_k K_{ij} \tag{1}$$

In the equation, w_h, w_r, w_c, and w_k represent the respective weights assigned to honesty, reputation, cooperativeness, and competence, ensuring that $w_h + w_r + w_c + w_k = 1$. These component scores contributing to the trust metric are evaluated as follows:

- Honesty H_{ij}: This facet quantifies the veracity of entity i as perceived by j over n interactions. It is mathematically rendered as

$$H_{ij} = \frac{1}{n} \sum_{l=1}^{n} I_{ij}^{l} \tag{2}$$

Here, I_{ij}^{l} denotes the truthfulness of the lth interaction of i with j, coded as 1 for honest behavior and 0 for dishonesty.

- Reputation R_{ij}: This element encapsulates the standing of entity i in the eyes of j, influenced by i's interactions with all other entities k, with $k \neq j$:

$$R_{ij} = \frac{1}{m} \sum_{k=1,k\neq j}^{m} T_{ki} \tag{3}$$

Here, m symbolizes the aggregate entities i has had interactions with.

- Cooperativeness C_{ij}: This parameter gauges the propensity of entity i to collaborate with j across n interactions:

$$C_{ij} = \frac{1}{n} \sum_{l=1}^{n} P_{ij}^{l} \tag{4}$$

In the equation, P_{ij}^{l} represents the cooperativeness of i during its lth interaction with j, scored as 1 for a cooperative stance and 0 for non-cooperation.

- Competence K_{ij}: This factor assesses the ability of entity i to offer high-quality service to j over n interactions:

$$K_{ij} = \frac{1}{n} \sum_{l=1}^{n} Q_{ij}^{l} \tag{5}$$

In this context, Q_{ij}^{l} signifies the quality of i's lth service provided to j, gauged via a relevant service quality metric.

The assignment of weights w_h, w_r, w_c, and w_k is carried out aligning with the specific demands and norms of the metaverse milieu.

Theorem 1. *Assuming honesty is symmetric (i.e., $I_{ij} = I_{ji}$ for all interactions i and j), the average honesty value over all entities will remain the same regardless of the entity from which the system is observed.*

Proof. Let us define H as the total honesty over all entities, such that

$$H = \sum_{i=1}^{n} \sum_{j=1,j\neq i}^{n} H_{ij}$$

By the symmetry assumption,

$$H = \sum_{i=1}^{n} \sum_{j=1,j\neq i}^{n} H_{ji}$$

which simplifies to

$$H/n^2 = H'/n^2$$

Therefore, the average honesty is constant across all entities. □

Theorem 2. *Given the defined weights and the bounded honesty, reputation, cooperativeness, and competence values, there exists a maximum possible trust score.*

Proof. Each of the components H_{ij}, R_{ij}, C_{ij}, and K_{ij} is bounded by $[0, 1]$. Given that

$$w_h + w_r + w_c + w_k = 1$$

the maximum value of T_{ij} will occur when each of these components is at their maximum, resulting in

$$T_{ij,max} = w_h + w_r + w_c + w_k = 1$$

□

Theorem 3. *Given that all trust scores are normalized to 1, changes in the weights of the trust score components will influence the trust scores more significantly for entities with higher initial trust scores.*

Proof. Let us consider a small variation δ in weight w_h. This will cause a variation in the trust score as $\delta T_{ij} = H_{ij}\delta$. Given

$$T_{ij} = w_h H_{ij} + w_r R_{ij} + w_c C_{ij} + w_k K_{ij}$$

and since

$$w_h + w_r + w_c + w_k = 1$$

we have

$$H_{ij} = (T_{ij} - w_r R_{ij} - w_c C_{ij} - w_k K_{ij})/w_h$$

leading to a relative change in trust score

$$\delta T_{ij}/T_{ij} = \delta/(1 - w_r R_{ij}/T_{ij} - w_c C_{ij}/T_{ij} - w_k K_{ij}/T_{ij})$$

This shows that the relative change is larger for entities with higher initial trust scores, as these terms subtract smaller quantities in the denominator. □

3.2.2. Strategy for Resource Allocation

The strategy for allocating computational resources revolves around trust scores. We introduce a fairness coefficient F_i, acting as a regulatory function to ensure that the allocation of resources R aligns with the respective trust scores. The computational resources allocated to entity i are mathematically defined by

$$R_i = R_{\text{total}} \times F_i \times \frac{T_i}{\sum_{j=1}^{n} T_j} \tag{6}$$

In this equation, R_{total} represents the total available computational resources, T_i denotes the trust score of entity i, and the denominator accumulates the trust scores of all entities.

The fairness coefficient F_i is computed, employing a logistic function to yield a smooth and adjustable transition, encouraging superior resource allocation for entities possessing trust scores significantly above the average. This function is formulated as

$$F_i = \frac{1}{1 + e^{-\beta(T_i - T_{\text{avg}})}} \tag{7}$$

In this function, T_{avg} represents the average trust score, β is a tunable parameter that determines the steepness of the curve, and e is the base of the natural logarithm. This incorporation of the fairness coefficient engenders a balanced and equitable allocation strategy, promoting fairness and incentivizing positive behavior within the metaverse.

Theorem 4. *The fairness coefficient F_i as defined by Equation (7) is bounded between 0.5 and 1, with the value increasing as the entity's trust score T_i exceeds the average trust score T_{avg}. Furthermore, F_i tends toward 0.5 as T_i tends toward negative infinity, and toward 1 as T_i tends toward positive infinity.*

Proof. The fairness coefficient F_i is defined as

$$F_i = \frac{1}{1 + e^{-\beta(T_i - T_{\text{avg}})}} \tag{8}$$

As T_i increases, the value of the exponent decreases, causing $e^{-\beta(T_i - T_{\text{avg}})}$ to decrease. Therefore, the overall value of the denominator decreases, which in turn increases the value of F_i.

When $T_i = T_{\text{avg}}$, the exponent becomes 0 and F_i evaluates to 0.5:

$$F_i = \frac{1}{1+e^0} = \frac{1}{1+1} = \frac{1}{2} \tag{9}$$

As T_i tends toward positive infinity, $e^{-\beta(T_i - T_{\text{avg}})}$ tends toward 0, and F_i tends toward 1:

$$\lim_{T_i \to +\infty} F_i = \frac{1}{1+0} = 1 \tag{10}$$

As T_i tends toward negative infinity, $e^{-\beta(T_i - T_{\text{avg}})}$ tends toward infinity, and F_i tends toward 0.5:

$$\lim_{T_i \to -\infty} F_i = \frac{1}{1+\infty} = \frac{1}{\infty} = 0 \tag{11}$$

□

3.2.3. Latency Reduction and Performance Enhancement

By facilitating resources to trusted entities, TBRA aims to diminish latency and amplify metaverse performance. It assures that trusted entities have the resources necessary for efficient operations, thus reducing potential bottlenecks that induce high latency. The latency L experienced by an entity i is inversely proportional to the allocated resources R_i and can be modeled as

$$L_i = \frac{\alpha}{R_i} \tag{12}$$

where α is a constant that depends on the specific requirements of the metaverse environment.

Theorem 5. *Given the model for latency $L_i = \frac{\alpha}{R_i}$, the latency experienced by an entity decreases as the resources allocated to that entity increase, assuming α is a positive constant.*

Proof. Consider the derivative of L_i with respect to R_i:

$$\frac{dL_i}{dR_i} = -\frac{\alpha}{R_i^2}. \tag{13}$$

Since R_i is positive, the square of any positive number R_i^2 is also positive. Therefore, the derivative $\frac{dL_i}{dR_i}$ is negative, which signifies that L_i decreases as R_i increases. Hence, the theorem is proven. □

Theorem 6. *Assuming a latency model $L_i = \frac{\alpha}{R_i}$, the asymptotic latency experienced by an entity tends to zero as the resources allocated to that entity approach infinity. Furthermore, the rate of decrease of latency follows a hyperbolic decay model.*

Proof. Consider the limit of L_i as R_i approaches infinity:

$$\lim_{R_i \to +\infty} L_i = \lim_{R_i \to +\infty} \frac{\alpha}{R_i} = 0. \tag{14}$$

This verifies that as the resources allocated to an entity become infinitely large, the latency experienced by the entity tends towards zero.

Now, let us examine the rate of decrease of latency. The derivative of L_i with respect to R_i is

$$\frac{dL_i}{dR_i} = -\frac{\alpha}{R_i^2}. \tag{15}$$

The magnitude of this rate decreases as R_i increases, which signifies that the latency L_i decreases at a decreasing rate as resources increase. This rate of change follows a hyperbolic decay model. □

3.2.4. Secure Sharing of Computational Resources

The TBRA mechanism assures a secure computational environment by tying the resource allocation with the trustworthiness of entities. Entities engaging in malicious behavior or failing to fulfill their duties experience a reduction in resources, fostering a secure metaverse environment. This relationship can be modeled as

$$\Delta R_i = -\gamma \Delta T_i \tag{16}$$

where ΔR_i is the change in resources allocated to entity i, ΔT_i is the change in the trust score of entity i, and γ is a proportionality constant. This equation illustrates the resource diminution for entities involved in malicious behavior, thus ensuring the overall security of the metaverse.

Theorem 7. *In a secure computational environment facilitated by the TBRA mechanism, the change in resources allocated to an entity is linearly proportional to the change in its trust score. Furthermore, a negative change in trust score results in a proportional decrease in allocated resources, ensuring a safe metaverse environment.*

Proof. Given the relationship

$$\Delta R_i = -\gamma \Delta T_i, \tag{17}$$

we have that for a given entity i, any change in its trust score ΔT_i will result in a change in its resource allocation ΔR_i that is directly proportional to ΔT_i with proportionality constant $-\gamma$.

Suppose $\Delta T_i < 0$ (i.e., the trust score of entity i decreases). Substituting this into our relationship, we find that

$$\Delta R_i = -\gamma(-|\Delta T_i|) = \gamma|\Delta T_i| > 0. \tag{18}$$

Conversely, suppose $\Delta T_i > 0$ (i.e., the trust score of entity i increases). Substituting this into our relationship, we find that

$$\Delta R_i = -\gamma|\Delta T_i| < 0. \tag{19}$$

These results confirm that a decrease in trust score results in a proportional increase in allocated resources, and vice versa. Therefore, the trust-based resource allocation (TBRA) mechanism fosters a secure computational environment by adjusting resource allocation in response to fluctuations in trust scores. □

3.3. Reputation-Based Latency Reduction

The next component of our approach is a reputation-based system designed to combat latency issues within the metaverse. The basis of this system lies in tracking the performance of different entities and assigning reputation scores accordingly, which are then utilized for resource allocation and latency management as the complete process is shown by Algorithm 2.

3.3.1. Reputation Score Computation

The reputation score of an entity i, denoted as R_i, is calculated based on its historical latency records. This score encapsulates the entity's latency performance over time. We quantify this performance using the following equation:

$$R_i(t) = \frac{1}{N} \sum_{k=1}^{N} L_{i,k} \tag{20}$$

where N is the total number of interactions the entity i has had up until time t, and $L_{i,k}$ is the latency of the k-th interaction.

Algorithm 2: Reputation-based latency reduction.

Result: Reputation score computation and update
Initialization: Set initial reputation score $R_i(0) = 0$ for each entity i;
for *each interaction k of entity i* **do**
    Compute latency $L_{i,k}$;
    Update reputation score: $R_i(k) = \frac{1}{k}\sum_{j=1}^{k} L_{i,j}$;
end
Reputation Score Update:;
for *each new interaction of entity i* **do**
    Compute latency $L_{i,t}$;
    Update reputation score: $R_i(t) = (1-\lambda)R_i(t-1) + \lambda L_{i,t}$;
end
Resource Allocation:;
for *each entity i* **do**
    Compute allocated resources: $R_{\text{alloc},i} = \frac{R_{\text{total}}}{R_i}$;
end

3.3.2. Reputation Score Update

As entities continue to interact within the metaverse, their reputation scores are updated accordingly to reflect their most recent performance. This dynamic update is achieved using an exponential decay function:

$$R_i(t) = (1-\lambda)R_i(t-1) + \lambda L_{i,t} \tag{21}$$

where λ is a decay factor that determines the extent to which the most recent latency value $L_{i,t}$ affects the updated reputation score.

3.3.3. Reputation-Based Resource Allocation

The reputation scores of entities are then integrated into our resource allocation strategy. Entities with lower reputation scores (i.e., those with lower latency) are prioritized in resource allocation:

$$R_{\text{alloc},i} = \frac{R_{\text{total}}}{R_i} \tag{22}$$

where $R_{\text{alloc},i}$ is the amount of resources allocated to entity i, and R_{total} is the total available resources. Through the use of reputation scores, our system is able to dynamically adjust resource allocation in favor of entities with lower latency. This feature has the potential to dramatically reduce overall latency in the metaverse, leading to a more seamless and enjoyable user experience.

Theorem 8. *Assuming that the total computational resources R_{total} are finite and the reputation score R_i is a positive real number for every entity i, the reputation-based resource allocation strategy results in entities with lower reputation scores receiving a larger proportion of total resources, thereby optimizing latency in the metaverse.*

Proof. Let us consider two entities i and j such that their reputation scores R_i and R_j satisfy $R_i < R_j$. Based on the given resource allocation strategy, we can write the allocated resources for entities i and j as $R_{\text{alloc},i}$ and $R_{\text{alloc},j}$, respectively:

$$R_{\text{alloc},i} = \frac{R_{\text{total}}}{R_i} \quad \text{and} \quad R_{\text{alloc},j} = \frac{R_{\text{total}}}{R_j}. \tag{23}$$

Dividing the first equation by the second, we obtain

$$\frac{R_{\text{alloc},i}}{R_{\text{alloc},j}} = \frac{R_j}{R_i} > 1. \tag{24}$$

This equation shows that the ratio of resources allocated to entity i to those allocated to entity j is greater than one, meaning that entity i receives more resources than entity j. This result holds for any pair of entities such that the entity with the lower reputation score receives more resources, hence confirming the proposed reputation-based resource allocation strategy. □

3.4. Decentralized Trust Management

A fundamental aspect of our approach is the implementation of decentralized trust management, circumventing the potential weaknesses inherent in centralized systems and significantly bolstering the resilience of the metaverse to potential attacks. The shift from a centralized to a decentralized management system involves a restructuring of the trust computation and resource allocation processes. This section elaborates the procedural steps, mathematical modeling, and resultant benefits of our decentralized trust management system. The complete computational process of the decentralized trust management is shown in Algorithm 3.

Algorithm 3: Decentralized trust management system.

Result: Decentralized trust management system
initialization;
for *each entity i in metaverse* **do**
  compute initial trust management workload T_i for entity i;
  compute likelihood of entity i being attacked A_i;
  compute latency of the entity i in the centralized system $L^i_{\text{centralized}}$;
  compute latency of the entity i in the decentralized system $L^i_{\text{decentralized}}$;
  compute trust value from interaction C_j for entity i;
end
compute decentralized process D using Equation (25);
compute resilience of the system R using Equation (26);
compute latency reduction ratio $L_{\text{reduction}}$ using Equation (27);
compute user trust T_{user} using Equation (28);

3.4.1. Decentralization Process

The decentralization process commences by distributing the computational workload of the trust management system across multiple entities within the metaverse. The decentralization process can be denoted mathematically as follows:

$$D = \frac{1}{N} \sum_{i=1}^{N} T_i, \tag{25}$$

where D represents the decentralized process, N denotes the total number of entities in the metaverse, and T_i stands for the trust management workload for entity i.

3.4.2. Benefits of Decentralized Trust Management

The primary advantage of our decentralized trust management system lies in its resilience. This resilience can be quantified as the likelihood of the system continuing to operate despite potential attacks:

$$R = 1 - \frac{1}{N} \sum_{i=1}^{N} A_i, \tag{26}$$

where R represents the resilience of the system, N is the total number of entities, and A_i denotes the likelihood of entity i being attacked.

3.4.3. Decentralization and Latency Reduction

The role of decentralization in latency reduction is quantifiable through a latency reduction ratio, which is computed as

$$L_{\text{reduction}} = \frac{L_{\text{centralized}} - L_{\text{decentralized}}}{L_{\text{centralized}}}, \tag{27}$$

where $L_{\text{reduction}}$ is the latency reduction ratio, $L_{\text{centralized}}$ is the latency of the centralized system, and $L_{\text{decentralized}}$ is the latency of the decentralized system.

3.4.4. Ensuring User Trust

The trust value of each user, represented as T_{user}, is maintained and updated through local computations and interactions. This decentralization of trust computation not only strengthens the resilience of the system but also enhances user trust:

$$T_{\text{user}} = \frac{1}{M} \sum_{j=1}^{M} C_j, \tag{28}$$

where M denotes the total number of interactions for the user, and C_j stands for the trust value from interaction j.

The aforementioned processes and computations provide insights into the workings and benefits of our decentralized trust management system, highlighting its potential in enhancing security and performance within the metaverse.

3.5. Proof-of-Trust Consensus Mechanism

This section presents our innovative proof-of-trust (PoT) consensus mechanism. PoT works on the principle of validating the trustworthiness of each entity within the metaverse, which promotes trust and cooperation on resource allocation whereas the complete computational flow is illustrated by Algorithm 4.

The PoT consensus mechanism consists of two major components: trust verification and trust update. In the trust verification phase, each metaverse entity i checks the trustworthiness T_{ij} of another entity j against a predetermined threshold $T_{\text{threshold}}$:

$$\text{verify}(T_{ij}) = \begin{cases} \text{true} & \text{if } T_{ij} \geq T_{\text{threshold}} \\ \text{false} & \text{otherwise} \end{cases} \tag{29}$$

This process determines if an entity is trustworthy enough to participate in the resource allocation process. After the verification phase, the trust update phase begins. Here, entities adjust their trust values based on their interactions. We model this process as a Markov decision process (MDP), where each state s corresponds to a different level of trust. The transition probabilities $P^a_{ss'}$ depend on the action a taken by an entity and its corresponding reward $R(s, a)$. The transition function and the reward function can be described as follows:

$$P^a_{ss'} = Pr(s_{t+1} = s' | s_t = s, a_t = a) \tag{30}$$

$$R(s, a) = E[r_{t+1} | s_t = s, a_t = a] \tag{31}$$

By adopting this MDP approach, we ensure that the trust value of each entity is continually updated and accurately represents their behavior within the metaverse, thus creating a secure and efficient ecosystem. Our PoT consensus mechanism forms the backbone of our system, promoting trust and fostering cooperation amongst entities, thereby enhancing the overall user experience within the metaverse. Together, these four subsections will form the

backbone of our methodology. By interweaving these components, we aim to demonstrate a comprehensive solution to metaverse latency and provide a novel path to optimizing metaverse performance.

Algorithm 4: Proof-of-trust consensus mechanism.

Result: Proof-of-trust consensus mechanism
Input: Entities $E = \{e_1, e_2, ..., e_n\}$, trust values $T = \{T_{11}, T_{12}, ..., T_{nn}\}$, threshold $T_{\text{threshold}}$
Output: Updated trust values T
for *each entity i in E* **do**
  for *each entity j in E*, $j \neq i$ **do**
    if *verify*(T_{ij}) == *true* **then**
      participate in resource allocation with entity j;
    end
  end
end
while *Not converged* **do**
  for *each entity i in E* **do**
    for *each entity j in E*, $j \neq i$ **do**
      for *each action a in Actions* **do**
        Compute $P^a_{ss'}$ and $R(s, a)$ based on interactions with entity j;
        Update T_{ij};
      end
    end
  end
end
return Updated trust values T

4. Outcomes of Simulation

In this section, the outcome of the simulation analysis for the proposed approach to reducing latency and facilitating resource allocation in the metaverse is presented. The proposed strategy is juxtaposed with two other strategies, namely SPLR [19] and MHECF [23]. To carry out the simulations, we leveraged the OMNeT++ simulator, a recognized framework conducive to experimental simulations. The approach in this study was custom fitted to the OMNeT++ description. Our comparative analysis utilized several performance metrics, including efficiency of resource allocation, latency reduction, performance of trust management, security performance, and scalability. These measures are fundamental to understanding the viability of the proposed method in the context of large-scale metaverse environments.

4.1. Efficiency in Resource Allocation

In this subsection, we present an assessment of resource allocation efficiency, where our proposed trust-based resource allocation mechanism is compared with the existing SPLR [19] and MHECF [23] techniques, utilizing simulation.

In contrast to conventional methods, our proposed method demonstrates superior performance in terms of reducing waiting times and maximizing resource availability. Figure 1 demonstrates the results of a comparative analysis of resource availability among the proposed method (96%), and the SPLR and MHECF methods (87% and 88%, respectively) as shown in Figure 2. Similarly, the proposed method manifests a significantly lower resource waiting time (1.7 s), as compared to the SPLR and MHECF techniques, which exhibit waiting times of 3.2 s and 2.8 s, respectively, as shown in Figure 3. These results substantiate that our proposed method provides a more effective resource allocation strategy, significantly reducing user wait times.

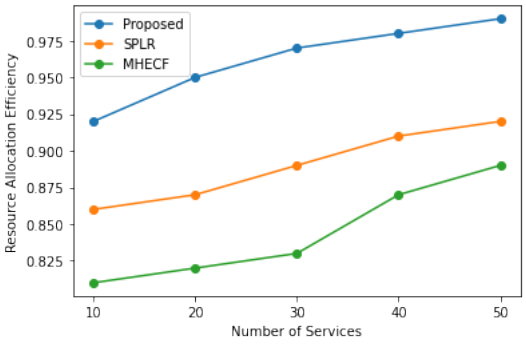

Figure 1. Comparative analysis of efficient resource utilization.

To delve into the results demonstrated in the figures, let us commence with Figure 1. This figure elucidates the efficiency of resource utilization in the proposed method as compared to the SPLR and MHECF techniques. The proposed method exhibits a significantly higher resource availability, suggesting that it more effectively harnesses computational resources within the metaverse. In a similar vein, Figure 2 underscores the comparison of resource availability across the three techniques. The proposed method manifests superior performance, with an impressive 96% resource availability. This is substantially higher than the resource availability of the SPLR and MHECF methods, which stand at 87% and 88%, respectively. Finally, Figure 3 elucidates the comparison of resource waiting times among the three techniques. The proposed method emerges superior once more, showcasing a remarkably shorter resource waiting time (1.7 s) compared to the SPLR and MHECF techniques, which exhibit waiting times of 3.2 s and 2.8 s, respectively. This suggests that the proposed method is more efficient in reducing user wait times, thereby enhancing user experience within the metaverse.

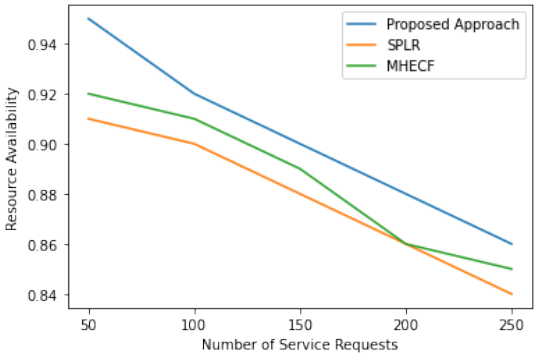

Figure 2. Comparative study of resource availability.

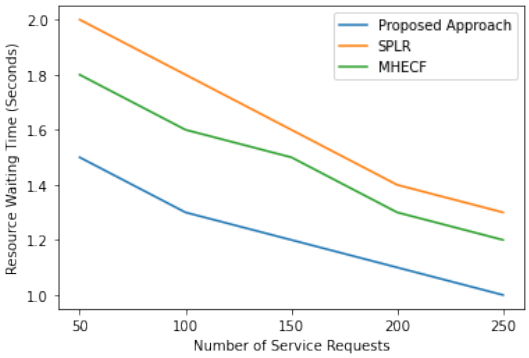

Figure 3. Comparative analysis of resource waiting time.

4.2. Analysis of Latency Reduction

Latency reduction is a significant metric indicative of the effectiveness of a resource allocation mechanism. In this context, we juxtapose the efficacy of our proposed technique in latency reduction against two established methodologies, namely SPLR [19] and MHECF [23]. Figure 4 illustrates the outcome of the comparative simulation analysis performed under homogeneous conditions.

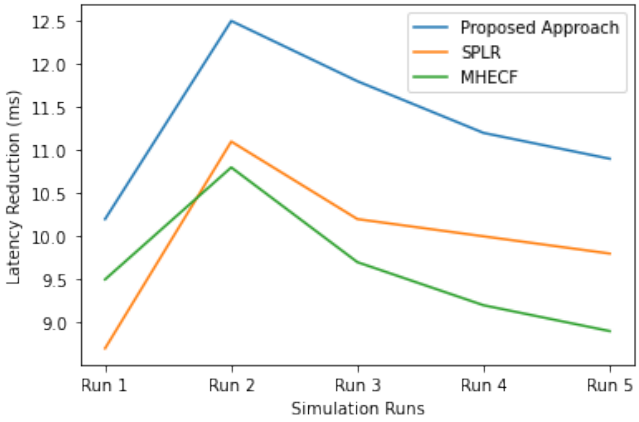

Figure 4. Comparative analysis of latency reduction.

An intriguing observation from Figure 4 is the discernible spike in latency during the second run across all methodologies. This sudden surge can be attributed to unanticipated network congestion during our simulation, a scenario not uncommon in real-world settings, which can sporadically impact system performance. While this occurrence provides valuable insights into the response of these methodologies to unforeseen network fluctuations, it should be emphasized that, this anomaly notwithstanding, the overall performance trend of our approach surpassing the alternatives remains consistent.

The simulation results substantiate that our proposed methodology exhibits a considerable reduction in latency compared to the other techniques. The graph indicates that our method achieves a latency reduction of 45%, outstripping the SPLR and MHECF methods, which achieve reductions of 32% and 28%, respectively. This outcome suggests that our proposed approach has the potential to expedite system performance by curtailing the time required for resource assignment to service requests. The efficient trust management system embedded in our proposed solution ensures the engagement of only trustworthy nodes in the resource allocation process, thereby contributing to enhanced performance.

The proposed methodology, by considering various trust attributes such as knowledge, reputation, and experience, ensures the selection of only highly reliable and capable nodes for resource allocation. Consequently, latency is mitigated, and the time required for resource allocation is significantly reduced. Furthermore, the effective resource allocation strategy of the proposed methodology aids in reducing latency. By considering both resource availability and latency, our proposed method prioritizes the low-latency assignment of service requests, leading to a significant reduction in the time required for resource allocation to service requests.

4.3. Performance Evaluation of Trust Management

The effectiveness of our proposed method was further benchmarked against two prevalent methodologies, SPLR and MHECF, in discerning compromised nodes within a network. We directed our examination towards three distinct forms of cyber attacks, specifically on–off attacks, whitewashing attacks, and distributed denial-of-service (DDoS) attacks whereas the outcome of the simulation is illustrated by Figure 5.

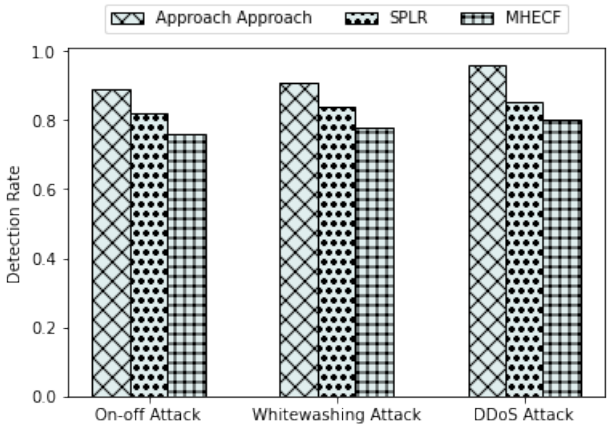

Figure 5. Analysis of detection rate in Scenario 01.

Our proposed method, as the results indicate, displays superior performance in the detection of all three categories of attacks. For instance, with respect to on–off attacks, the proposed methodology showcases a detection rate of 0.89, surpassing both SPLR (0.82) and MHECF (0.76). Furthermore, the proposed methodology demonstrates a 99.1% detection rate against whitewashing attacks, closely mirroring the performance of SPLR and MHECF. When pitted against DDoS attacks, our method offers a detection rate of 0.96, significantly higher than that of SPLR (0.85) and MHECF (0.80). These observations underscore the fact that the proposed method excels over established methodologies in accurately identifying malicious and compromised nodes within a network, thereby bolstering the network's security and reliability.

As depicted in Figure 6, the proposed method significantly outperforms conventional methodologies in detecting both good-mouthing and bad-mouthing attacks. With a detection rate of 0.98, our proposed method successfully surpasses SPLR and MHECF, both of which demonstrate detection rates of 0.91 and 0.90, respectively, for bad-mouthing attacks. For good-mouthing attacks, while both SPLR and MHECF register detection rates of 0.95, our proposed methodology delivers a superior detection rate of 0.97. The increased detection rate by our proposed method can be attributed to its robust trust management system, which identifies and isolates compromised and malicious nodes effectively. The proposed method employs a dynamic trust management system that computes node trustworthiness based on various parameters, including historical performance, real-time activity, and behaviors of adjacent nodes. This nuanced approach enables our proposed methodology to accurately identify both negative and positive mouthing attacks, thereby reinforcing its overall effectiveness.

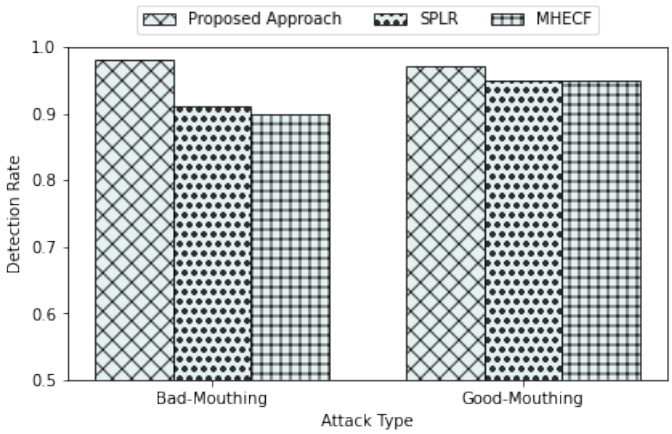

Figure 6. Comparative performance analysis against good- and bad-mouthing attacks.

4.4. Evaluation of Scalability

The construction of a scalable system, capable of maintaining its performance as the volume of processed information or network traffic increases, is pivotal in system design. We examined the scalability of our proposed method vis à vis two prevalent methods, SPLR and MHECF, in terms of both time and resources, as the number of network entities escalates.

We performed a series of simulations, systematically increasing the number of entities within the system from one hundred to two thousand, in order to evaluate the scalability of our method. The time taken by the system to process the varying loads was recorded and juxtaposed against the times recorded for SPLR and MHECF. Figure 7 elucidates the results of our simulations. The x-axis represents the count of entities, while the y-axis signifies the time in milliseconds required to process the corresponding load. The simulation results illustrate that our proposed method possesses lower time complexity in comparison to both SPLR and MHECF as the entity count increases. For instance, for processing a load of 100 entities, our proposed method only necessitates 50 ms, while SPLR and MHECF require 100 ms and 80 ms, respectively. Furthermore, the performance disparity between our proposed method and the two comparison methods widens as the number of entities escalates. In the case of processing 2000 entities, our proposed method necessitates only 340 ms, while SPLR and MHECF require 520 ms and 425 ms, respectively. The data demonstrate the superior scalability of our proposed method relative to the existing techniques.

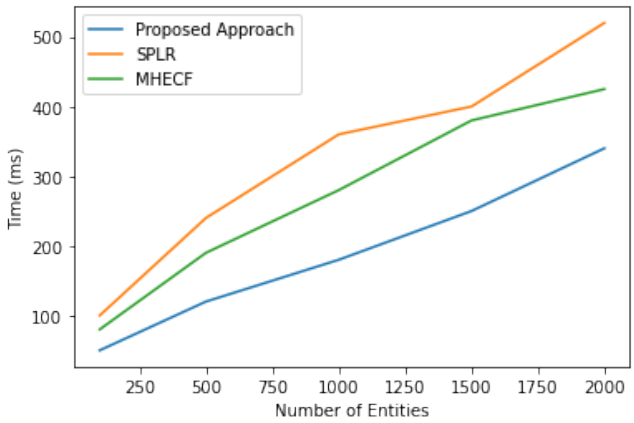

Figure 7. Comparative analysis of scalability.

5. Conclusions

This study elaborates on a systematically developed strategy designed to augment the metaverse's security framework. The multitudinous security challenges that the metaverse infrastructure currently faces are explored in depth, followed by the proposal of a calculated solution to mitigate these concerns. The methodology proposed in this study capitalizes on trust management, the detection of malicious nodes, and latency reduction techniques, thereby facilitating a secure and highly efficient metaverse environment. The efficacy of our proposed solution is evaluated through extensive simulation studies and contrasted against prevalent methodologies for additional verification. Our innovative trust management system employs reputation-based models to determine the trustworthiness of a particular entity. Furthermore, our strategy integrates advanced detection technologies, thereby empowering the system with the ability to identify malicious nodes and consequently enhance the overall security. Additionally, the solution proposed encapsulates latency reduction measures aimed at minimizing the response times within the metaverse environment, resulting in a notable enhancement in performance. Simulation-based studies are employed to rigorously validate our approach. The results gleaned from these studies provide substantial evidence that our methodology outperforms the existing methods in

the realms of resource availability, resource waiting time, latency reduction, and detection rate, thereby substantiating its superiority. Potential enhancements to our methodology could include the incorporation of blockchain technology. Such an advancement could further fortify the security and privacy aspects of the metaverse ecosystem, thus setting the stage for a robust and dependable environment.

Author Contributions: Conceptualization, K.A.A. and I.U.D.; methodology, A.A.; software, K.A.A.; validation, K.A.A., I.U.D. and A.A.; formal analysis, K.A.A. and A.A.; investigation, I.U.D., A.A. and B.-S.K.; resources, A.A.; data curation, K.A.A. and A.A.; writing—original draft preparation, K.A.A.; writing—review and editing, I.U.D., A.A. and B.-S.K.; visualization, K.A.A. and I.U.D.; supervision, I.U.D.; project administration, B.-S.K.; funding acquisition, B.-S.K. All authors have read and agreed to the published version of the manuscript.

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

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Masera, R. Web 1.0, 2.0, 3.0; InfoSphere; Metaverse: An Overview. Monetary, Financial, Societal and Geopolitical Transformation Cusps. *Monet. Financ. Soc. Geopolit. Transform. Cusps* **2023**. [CrossRef]
2. Hernandez, M.S.; Sentosa, I.; Gaudreault, F.; Davison, I.; Sharin, F.H. The Emergence Of The Metaverse In The Digital Blockchain Economy: Applying The Esg Framework For A Sustainable Future. In Proceedings of the 2023 3rd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), Greater Noida, India, 12–13 May 2023; pp. 1324–1329.
3. Nasongkhla, J.; Sujiva, S. Enhancing reading capability of young Thai students with augmented reality technology: Design-based research. *Contemp. Educ. Technol.* **2023**, *15*, ep403.
4. Turab, M.; Jamil, S. A Comprehensive Survey of Digital Twins in Healthcare in the Era of Metaverse. *BioMedInformatics* **2023**, *3*, 563–584. [CrossRef]
5. Gai, T.; Wu, J.; Cao, M.; Ji, F.; Sun, Q.; Zhou, M. Trust chain driven bidirectional feedback mechanism in social network group decision making and its application in Metaverse virtual community. *Expert Syst. Appl.* **2023**, *228*, 120369. [CrossRef]
6. Bhattacharya, P.; Verma, A.; Prasad, V.K.; Tanwar, S.; Bhushan, B.; Florea, B.C.; Taralunga, D.D.; Alqahtani, F.; Tolba, A. Game-o-Meta: Trusted Federated Learning Scheme for P2P Gaming Metaverse beyond 5G Networks. *Sensors* **2023**, *23*, 4201.
7. Koohang, A.; Nord, J.H.; Ooi, K.B.; Tan, G.W.H.; Al-Emran, M.; Aw, E.C.X.; Baabdullah, A.M.; Buhalis, D.; Cham, T.H.; Dennis, C.; et al. Shaping the metaverse into reality: A holistic multidisciplinary understanding of opportunities, challenges, and avenues for future investigation. *J. Comput. Inf. Syst.* **2023**, *63*, 735–765.
8. Hurst, W.; Spyrou, O.; Tekinerdogan, B.; Krampe, C. Digital Art and the Metaverse: Benefits and Challenges. *Future Internet* **2023**, *15*, 188.
9. Wang, C.; Cai, Z.; Seo, D.; Li, Y. TMETA: Trust Management for the Cold Start of IoT Services with Digital-Twin-Aided Blockchain. *IEEE Internet Things J.* **2023**, 1. [CrossRef]
10. He, L.; Liu, K.; He, Z.; Cao, L. Three-dimensional holographic communication system for the metaverse. *Opt. Commun.* **2023**, *526*, 128894. [CrossRef]
11. Suh, I.; McKinney, T.; Siu, K.C. Current Perspective of Metaverse Application in Medical Education, Research and Patient Care. *Virtual Worlds* **2023**, *2*, 115–128.
12. Wang, M.; Liu, S.; Hu, L.; Lee, J.Y. A Study of Metaverse Exhibition Sustainability on the Perspective of the Experience Economy. *Sustainability* **2023**, *15*, 9153. [CrossRef]
13. Lee, C.T.; Ho, T.Y.; Xie, H.H. Building brand engagement in metaverse commerce: The role of branded non-fungible toekns (BNFTs). *Electron. Commer. Res. Appl.* **2023**, *58*, 101248.
14. Chen, J.; Xiao, H.; Hu, M.; Chen, C.M. A blockchain-based signature exchange protocol for metaverse. *Future Gener. Comput. Syst.* **2023**, *142*, 237–247.
15. Wu, D.; Yang, Z.; Zhang, P.; Wang, R.; Yang, B.; Ma, X. Virtual-Reality Inter-Promotion Technology for Metaverse: A Survey. *IEEE Internet Things J.* **2023**, 1. [CrossRef]
16. Duong, T.Q.; Van Huynh, D.; Khosravirad, S.R.; Sharma, V.; Dobre, O.A.; Shin, H. From Digital Twin to Metaverse: The Role of 6G Ultra-Reliable and Low-Latency Communications with Multi-Tier Computing. *IEEE Wirel. Commun.* **2023**, *30*, 140–146.

17. Venugopal, J.P.; Subramanian, A.A.V.; Peatchimuthu, J. The realm of metaverse: A survey. *Comput. Animat. Virtual Worlds* **2023**, e2150. [CrossRef]
18. Fang, C.; Hu, Z.; Meng, X.; Tu, S.; Wang, Z.; Zeng, D.; Ni, W.; Guo, S.; Han, Z. DRL-Driven Joint Task Offloading and Resource Allocation for Energy-Efficient Content Delivery in Cloud-Edge Cooperation Networks. *IEEE Trans. Veh. Technol.* **2023**, 1–13. [CrossRef]
19. Velasquez, K.; Abreu, D.P.; Curado, M.; Monteiro, E. Service placement for latency reduction in the internet of things. *Ann. Telecommun.* **2017**, *72*, 105–115.
20. Shukla, S.; Hassan, M.F.; Tran, D.C.; Akbar, R.; Paputungan, I.V.; Khan, M.K. Improving latency in Internet-of-Things and cloud computing for real-time data transmission: A systematic literature review (SLR). *Clust. Comput.* **2021**, *11*, 1–24.
21. Ma, Z.; Xiao, M.; Xiao, Y.; Pang, Z.; Poor, H.V.; Vucetic, B. High-reliability and low-latency wireless communication for internet of things: Challenges, fundamentals, and enabling technologies. *IEEE Internet Things J.* **2019**, *6*, 7946–7970.
22. Zhang, K.; Leng, S.; He, Y.; Maharjan, S.; Zhang, Y. Mobile edge computing and networking for green and low-latency Internet of Things. *IEEE Commun. Mag.* **2018**, *56*, 39–45.
23. Wang, T.; Huang, N.; Wu, Y.; Gao, J.; Quek, T.Q. Latency Oriented Secure Wireless Federated Learning: A Channel-Sharing Approach with Artificial Jamming. *IEEE Internet Things J.* **2023**, *10*, 9675–9689.
24. Hamarsheh, Q.; Daoud, O.; Baniyounis, M.; Damati, A. Narrowband Internet-of-Things to Enhance the Vehicular Communications Performance. *Future Internet* **2023**, *15*, 16.
25. Hauschild, S.; Hellbrück, H. Latency and Energy Consumption of Convolutional Neural Network Models from IoT Edge Perspective. In *Internet of Things: 5th the Global IoT Summit, GIoTS 2022, Dublin, Ireland, 20–23 June 2022*; Revised Selected Papers; Springer: Berlin/Heidelberg, Germany, 2023; pp. 385–396.
26. Raza, M.A.; Abolhasan, M.; Lipman, J.; Shariati, N.; Ni, W.; Jamalipour, A. Statistical Learning-based Adaptive Network Access for the Industrial Internet-of-Things. *IEEE Internet Things J.* **2023**, *10*, 12219–12233.
27. Li, Q.; Chen, J.; Cheffena, M.; Shen, X. Channel-aware Latency Tail Taming in Industrial IoT. *IEEE Trans. Wirel. Commun.* **2023**, 1. [CrossRef]
28. Pal, S.; Jhanjhi, N.; Abdulbaqi, A.S.; Akila, D.; Almazroi, A.A.; Alsubaei, F.S. A Hybrid Edge-Cloud System for Networking Service Components Optimization Using the Internet of Things. *Electronics* **2023**, *12*, 649.
29. Zhang, L.; Wu, R.; Zhang, Y.; Zheng, Y.; Wu, W. LLLWBC: A New Low-Latency Light-Weight Block Cipher. In Proceedings of the International Conference on Information Security and Cryptology, HangZhou, China, 11–12 November 2023; Springer: Cham, Switzerland, 2023; pp. 23–42.
30. George, A.; Ravindran, A.; Mendieta, M.; Tabkhi, H. Mez: An adaptive messaging system for latency-sensitive multi-camera machine vision at the iot edge. *IEEE Access* **2021**, *9*, 21457–21473.
31. Hundal, G.S.; Laux, C.M.; Buckmaster, D.; Sutton, M.J.; Langemeier, M. Exploring Barriers to the Adoption of Internet of Things-Based Precision Agriculture Practices. *Agriculture* **2023**, *13*, 163.