Computer Networks 254 (2024) 110803

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Survey paper

Survey of fault management techniques for edge-enabled distributed metaverse applications

Shahzaib Shaikh [*], Manar Jammal

School of Information Technology, York University, Toronto, Canada

ARTICLE INFO

Keywords:
Metaverse
Edge computing
Fault tolerance
Failure remediation
Distributed systems
Machine learning

ABSTRACT

The metaverse, envisioned as a vast, distributed virtual world, relies on edge computing for low-latency data processing. However, ensuring fault tolerance – the system's ability to handle failures – is critical for a seamless user experience. This paper analyzes existing research on fault tolerance in edge computing over the past six years, specifically focusing on its applicability to the metaverse. We identify common fault types like node failures, communication disruptions, and security issues. The analysis then explores various fault management techniques including proactive monitoring, resource optimization, task scheduling, workload migration, redundancy for service continuity, machine learning for predictive maintenance, and consensus algorithms to guarantee data integrity. While these techniques hold promise, adaptations are necessary to address the metaverse's real-time interaction requirements and low-latency constraints. This paper analyzes existing research and identifies key areas for improvement, providing valuable research guidelines and insights to pave the way for the development of fault management techniques specifically tailored to the metaverse, ultimately contributing to a robust and secure virtual world.

1. Introduction

The metaverse promises to redefine human interaction and digital experiences by envisioning a vast, immersive, and interconnected virtual world accessible through diverse devices. However, unlike traditional online games or virtual reality (VR) experiences, the metaverse aspires to be a distributed entity, free from a central server or governing authority. This distributed nature presents unique challenges in its realization.

One critical challenge is ensuring low-latency data processing. Latency, the time it takes for data to travel between devices, can significantly impact user experience in the metaverse. Delayed actions or a choppy virtual environment can be highly disruptive and can cause motion sickness. Security also presents a major concern. A distributed metaverse, with its numerous interconnected devices, expands the attack surface for malicious actors [1]. Protecting user data and maintaining system integrity become paramount concerns.

Moreover, the metaverse demands substantial computational resources to accommodate millions of concurrent users. The limitations in processing power and storage capacity of individual devices make achieving such scalability challenging.

Edge computing, a distributed computing paradigm, emerges as a promising enabling technology to tackle these challenges. It utilizes geographically distributed devices for processing at the network's edge [2], bringing computation and data storage closer to where it is needed. This improves response times and saves bandwidth by reducing latency, minimizing data travel distance, and lessening reliance on cloud servers. In the context of the metaverse, edge computing is crucial as it enables the low-latency processing necessary for real-time interactions within virtual environments.

To enhance security, edge computing localizes data processing and storage, reducing the need to transmit sensitive data over long distances to centralized cloud servers. This localization minimizes the potential exposure to cyber-attacks during data transmission. Furthermore, edge devices can implement localized security measures tailored to specific environments, allowing for more granular control over data access and integrity. While modern cloud data centers are logically centralized and incorporate redundancy measures to tolerate certain failures, edge computing's distributed nature further compartmentalizes data and processing tasks, potentially reducing the risk of a large-scale failure or breach affecting the entire system. Although edge computing introduces challenges related to managing security across multiple devices, its distributed nature generally offers enhanced protection compared to centralized cloud systems. Fig. 1 illustrates an edge-enabled metaverse architecture.

While edge computing significantly enhances the metaverse experience through low latency and high bandwidth, it is important

[*] Corresponding author.
E-mail address: shaikh98@yorku.ca (S. Shaikh).

https://doi.org/10.1016/j.comnet.2024.110803
Received 27 June 2024; Received in revised form 9 September 2024; Accepted 11 September 2024
Available online 13 September 2024

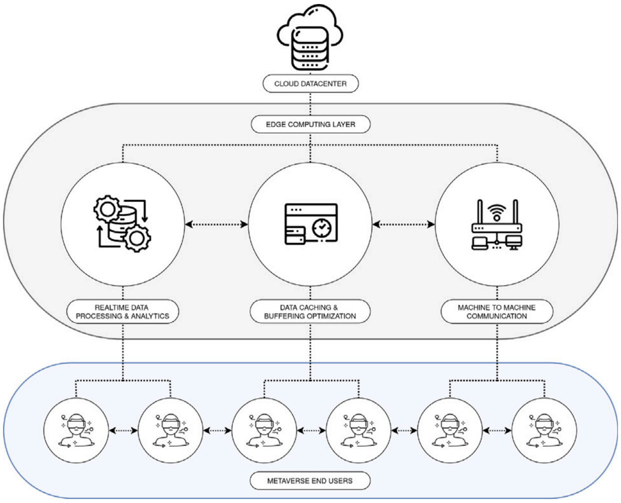

CLOUD DATACENTER
EDGE COMPUTING LAYER
REALTIME DATA PROCESSING & ANALYTICS
DATA CACHING & BUFFERING OPTIMIZATION
MACHINE TO MACHINE COMMUNICATION
METAVERSE END USERS

Fig. 1. Architecture for edge-enabled metaverse.

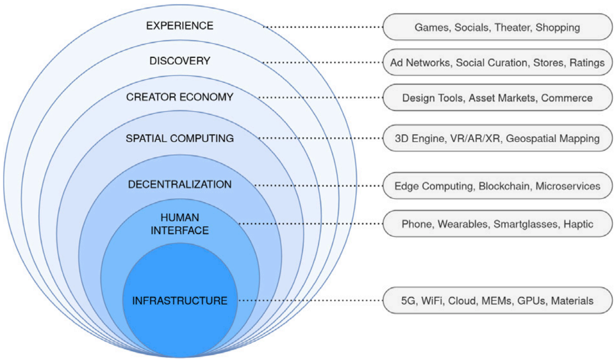

EXPERIENCE
Games, Socials, Theater, Shopping
DISCOVERY
Ad Networks, Social Curation, Stores, Ratings
CREATOR ECONOMY
Design Tools, Asset Markets, Commerce
SPATIAL COMPUTING
3D Engine, VR/AR/XR, Geospatial Mapping
DECENTRALIZATION
Edge Computing, Blockchain, Microservices
HUMAN INTERFACE
Phone, Wearables, Smartglasses, Haptic
INFRASTRUCTURE
5G, WiFi, Cloud, MEMs, GPUs, Materials

Fig. 2. Layers of metaverse.

to recognize that the metaverse could theoretically exist without it. However, such a scenario would likely face substantial challenges, including increased latency, reduced quality of service, and potential disruptions due to the centralized nature of cloud computing. These limitations underscore the critical role of edge computing in optimizing metaverse performance and scalability.

Despite the benefits of edge computing, the metaverse introduces fault management challenges that are distinct from those encountered in traditional computing environments. The dynamic and immersive nature of the metaverse requires fault management solutions that can quickly adapt to and mitigate disruptions. Unlike static systems, the metaverse's real-time interactions and immersive experiences demand rapid and effective fault detection and recovery mechanisms to ensure a consistent user experience.

Furthermore, the management of virtual assets distributed across a vast network adds complexity to fault management. The metaverse requires solutions capable of handling the distribution and synchronization of these assets, ensuring their integrity and availability across diverse, geographically dispersed nodes. The need to support a massive number of concurrent users and facilitate seamless interactions between different virtual environments and applications introduces additional scalability and interoperability challenges. Recent incidents, such as the high-profile server outages experienced by leading gaming platforms and virtual reality services like Xbox Live [3] and Steam [4], highlight the critical need for robust fault detection and recovery mechanisms in edge computing and metaverse applications.

In this paper, we conduct an empirical analysis of existing research on fault tolerance in edge computing over the last 6 years, focusing on their relevance to the distributed metaverse. Fault tolerance refers to the ability of a system to continue operation in the event of failures, ensuring uninterrupted service delivery. Our analysis has four main contributions:

1. We identify and analyze critical fault modes that can disrupt edge computing operations and degrade metaverse environments, assessing their impact on user experience and system functionality.
2. We conduct a rigorous examination of current research on fault remediation techniques employed in edge computing over the last six years.
3. We assess the effectiveness of existing techniques in addressing the unique challenges faced by edge computing when enabling metaverse applications.
4. We identify promising research avenues and explore potential solutions to address the unique failure tolerance requirements of the metaverse.

The remainder of this paper is organized as follows. Section 2 outlines the methodology used for empirical analysis. Section 3 examines the characterization of faults in edge computing environments. Section 4 critically evaluates existing research on fault remediation techniques for edge computing. Section 5 analyzes the applicability of these techniques in distributed metaverse applications. Section 6 presents a few case studies to demonstrate how specific challenges in the metaverse can be resolved by edge-based fault management techniques. Section 7 investigates gaps in literature and suggests future research directions. Finally, Section 8 concludes by summarizing key findings.

2. Methodology

In this section, we define the considered architectural models, scope of the survey and research questions for the conducted study.

2.1. Edge computing architectures

Within the paradigm of edge computing, several prominent architectures have emerged, each offering distinct advantages and deployment strategies:

(1) *Cloudlet* [5]: Introduces geographically dispersed micro-data centers, termed cloudlets, positioned at the network edge near users. Cloudlets connect via Wireless Local Area Networks (WLANs) and function as localized resource providers, offering computing and storage capabilities for reduced latency.
(2) *Multi-access Edge Computing (MEC)* [6]: Prioritizes low latency by integrating edge functionalities directly within the Radio Access Network (RAN) infrastructure, enabling faster processing at the network's edge.
(3) *Fog computing* [7]: Employs a decentralized approach, strategically deploying Fog Computing Nodes (FCNs) between the cloud and users. These FCNs offer broader reach compared to MEC model.

2.2. Metaverse components

The metaverse operates through a layered architecture, depicted in Fig. 2. One crucial layer is the decentralization layer, which heavily relies on edge computing to achieve low latency essential for seamless user experiences. Key functions performed atop the decentralization layer include:

- *Spatial Computing:* Involves creating and managing 3D virtual spaces. Edge computing supports real-time processing of spatial data at the network edge, minimizing latency for smooth user experiences.
- *Creator Economy:* Facilitates creation and customization of user-generated content. Edge computing ensures reliable data transfer, enabling creators to seamlessly design and interact with their content in a responsive spatial environment.
- *Discovery & Experience:* Essential for efficient content discovery and user experience. Edge computing enhances interactions and content searches within the metaverse, optimizing data processing and delivery.

2.3. Impact of edge failures on metaverse

Failures within the decentralization layer can have cascading effects, significantly impacting all layers built atop it. This layer is crucial for maintaining seamless connectivity, data integrity, and real-time processing. Disruptions here can propagate upward, leading to widespread issues such as:

- *Disrupted Spatial Awareness & Fidelity:* Users may encounter jittery or incorrectly placed objects, resulting in a disorienting and unrealistic experience. Delayed rendering can cause visual lag and stuttering, hindering immersion and potentially leading to motion sickness.
- *Inconsistencies in User Experience:* These can manifest as asynchronous user interactions, where actions performed by one user are not reflected instantaneously for others, creating a sense of disjointedness and frustration.
- *Hindered Real-time Communication:* Delays and disruptions in communication can severely impact social interactions within the metaverse. Users may experience lags in voice and video communication, disrupting the natural flow of social interactions.

2.4. Scope of the survey

To gather papers related to faults and failures in edge computing architectures, we utilized specific keywords in the Google Scholar search engine:

```
(fault OR failure OR high availability) AND
(tolerance OR remediation OR reliability OR recovery
OR redundancy OR checkpointing) AND (edge computing
OR fog computing OR cloudlets)
```

These keywords were carefully chosen to comprehensively capture research on fault and failure management within edge computing architectures. Concepts such as “fault” and “failure” address system malfunctions and outages, while terms like “tolerance”, “remediation”, “reliability”, “recovery”, and “checkpointing” cover mechanisms for prevention, detection, and recovery.

Given the extensive search results retrieved from the above query, we employed a two-step filtering approach. Firstly, we focused on research published between January 2019 and June 2024. This timeframe ensures coverage of recent advancements in edge computing, a rapidly evolving field. Secondly, we prioritized literature from only the most renowned conferences and journals listed in Table 1, ensuring the inclusion of high-quality studies on fault tolerance mechanisms.

Table 1
Surveyed journals and conferences.

Publisher	Conference / Journal
ACM	MobiCom, EdgeSys, SoCC, CitiFog
IEEE	EDGE, Cloud, ICFEC, FMEC, ICFC, CloudCom, IoT Journal
USENIX	CloudSum, DEBS, HotEdge, FOCI
Elsevier	Future Generation Computer Systems, Pervasive and Mobile Computing
Springer	Computing Journal, Complex & Intelligent Systems Journal, ICWS, NPC

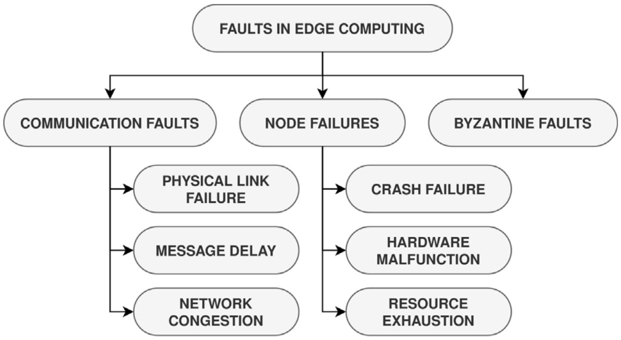


Fig. 3. Classification of faults & failures in edge computing.

2.5. Selection criteria

We limited our selection to research literature where the search keywords appeared directly within the title. Additionally, we excluded application-specific papers, particularly those focusing on domains with different fault tolerance requirements than those of the metaverse. For instance, research on fault tolerance in logistics management, though relevant to edge computing, often prioritizes real-time data tracking over the immersive experiences central to the metaverse. Similarly, fault tolerance in wearable health monitoring emphasizes data security and reliability over the low latency interactions essential for a seamless metaverse experience.

Instead, we prioritized research directly contributing to the metaverse, such as decentralized rendering, blockchain-based security protocols, and spatial computing. We also included studies on applications similar to the metaverse, like large-scale multiplayer online games (MMOs) and virtual reality (VR) collaboration platforms.

This rigorous selection process resulted in 45 publications that address fault tolerance challenges within edge computing environments. The subsequent sections of this paper analyze these findings and evaluate their relevance to metaverse applications.

3. Relevant fault types in edge computing

The successful operation of distributed metaverse applications hinges on the ability of underlying edge computing architectures to withstand various fault types. These faults, as categorized in Fig. 3, can originate from security issues, node failures or communication related faults, and can significantly disrupt system functionality.

3.1. Node failures

These consist of hardware malfunctions, operating system crashes, failure of virtual machines, processes, and edge devices. Node failures are more frequent in edge environments when compared to traditional cloud environments due its unique features. Unlike the controlled

Table 2
Fault types addressed in literature.

Fault Type	References	Count
Node Failures	[15–45]	31
Communication Faults	[46–53]	8
Byzantine Faults	[54–59]	6

environments within data centers, edge devices are deployed in diverse and often unpredictable locations. This exposes them to a higher risk of accidental damage and power outages [8]. Additionally, the inherent limitations of edge resources, including processing power and device heterogeneity, necessitate careful management to avoid resource exhaustion and subsequent crashes [9]. Furthermore, the reliability degradation of IoT edge devices also plays a significant role in causing node failures [10].

Node failures in the metaverse can manifest as users experiencing sudden disconnections, lagging interactions, or inconsistencies within the virtual environment. For example, if an edge device responsible for rendering a specific area of the metaverse malfunctions, users in that area might encounter visual glitches.

3.2. Communication faults

Edge systems rely on seamless data exchange between geographically dispersed devices and network resources. Disruptions in this communication can significantly impeding system functionality. Physical link failures, caused by faulty communication mediums or network equipment malfunctions, can completely halt data transmission [11]. Network congestion, due to increasing device density and data volume, can increase latency and cause packet loss. Additionally, service invocation failures, signal fluctuations, and varying network latencies can disrupt communication, leading to data loss.

In the metaverse, consequences of communication faults cascade throughout the system. Message loss disrupts data flow between edge devices, cloud resources, and IoT devices. Ultimately, these faults can degrade immersion and quality of service [12]. Severe and persistent faults may even result in complete user expulsion from the virtual environment, leading to frustration and diminished engagement.

3.3. Byzantine faults

Byzantine faults represent a challenging class of failures in distributed systems, encompassing unpredictable and malicious behavior beyond simple node failures [13]. A Byzantine faulty node can crash, send incorrect data, or actively disrupt the system by manipulating messages, making detection and consensus difficult within the system [14].

In the metaverse, a Byzantine faulty node can cause disruptions by transmitting erroneous data about avatars, leading to inconsistencies and breakdowns in spatial coherence. Furthermore, Byzantine faults could be exploited by malicious actors to manipulate messages and spread misinformation within the metaverse. This could destabilize social interactions and erode user trust in the platform.

In the 45 publications we reviewed in this survey, majority of the authors explored types of node failures and their remediation methods as summarized in Table 2.

4. Fault management techniques

Despite the inherent challenges posed by node failures and performance degradation faults, several fault management techniques have been proposed in the surveyed literature that can be employed to improve the reliability and resilience of edge computing architectures. These remediation techniques can be largely classified into the following categories based on their most prominent feature or approach. It is important to note that this classification does not aim to partition the solution space along a single dimension, but rather summarizes the solutions based on their most distinguishing aspect. Many techniques often overlap in terms of how they address faults and what they require, thus the classification aims to highlight the primary focus or contribution of each approach rather than exhaustively categorize them.

4.1. Proactive monitoring and resource optimization

Continuous monitoring of resource utilization on edge devices forms the foundation of fault management for performance-related issues. Real-time data enables proactive identification of potential bottlenecks and informed decision-making for resource optimization, specifically targeting performance-related faults that could disrupt the system. By focusing on performance issues, this approach helps to prevent faults that may arise from resource constraints, ensuring the smooth operation of edge environments.

Jia et al. [29] proposes a causal logging-based backup solution for stateful microservices deployed on edge devices in case of node failures. In a similar vein, authors in [28] utilize formal methods to find optimal servers for container redeployment in case of server crash.

Further studies have explored monitoring-based fault management solutions for resource exhaustion and high task queue lengths. Wang et al. [60] combine client-side workload reduction with server-side resource allocation to offload tasks based on application needs. Liu et al. [45] analyze rare high queue length events and optimize resource allocation using extreme value theory. Thantharate et al. [61] address the challenges of monitoring complex environments, such as DevOps, using Intelligent-Monitor. This approach improves real-time data handling and system visibility, which can be beneficial for proactive monitoring in edge computing.

Proactive monitoring and resource optimization rely on real-time data to identify bottlenecks and optimize resource allocation on edge devices. Its effectiveness in the metaverse depends on data specificity and monitoring data. Resource-intensive applications like collaborative design spaces may require more sophisticated monitoring and dynamic resource allocation compared to social interaction areas. Thus, the techniques proposed in the literature need adaptation to meet the specific needs of metaverse applications.

4.2. Task scheduling and workload migration

Task scheduling and workload migration are essential for addressing resource constraints and preventing performance bottlenecks in edge computing.

Focusing on communication failures, Lei et al. [46] proposes a heuristic-based service binding algorithm to achieve fault tolerance in MEC deployments with microservices, utilizing graphs and historical data for decision-making. Hou et al. [53] introduces the LR-UMEN framework, which optimizes communications, computing, and caching resources to minimize latency and enhance reliability.

Targeting node failures, Cheng et al. [20] presents a two-stage adaptive robust optimization model for resilient service placement and workload allocation in edge computing environments. Similarly, Liu et al. [16] enhances resilience through peer-collaborated service migration, moving services to other nodes within the edge cluster when failures occur or resources are overburdened.

Qiu et al. [62] propose a coded DEC framework with the SFPD code to handle large-scale tasks and mitigate latency issues. Their work is relevant for optimizing task scheduling and workload migration in edge computing, particularly for computation-intensive applications.

Task scheduling and migration distribute processing tasks across the edge network, crucial for the metaverse where user activity and virtual environment complexity cause fluctuating resource demands. However, the effectiveness of this strategy depends on data locality; tasks requiring frequent access to user-specific data may benefit less than those with lower data dependency. Despite the added complexity, consistent performance and user experience necessitate these techniques in the metaverse.

4.3. Redundancy and replication

Introducing redundancy involves deploying spare hardware components, replicating critical data across multiple edge devices, or utilizing geographically distributed edge servers for backup during localized failures.

Martinez et al. [27] employs a fault-tolerant fog architecture using standby fog nodes that activate upon detecting a nearby primary fog node failure. These standby nodes regularly synchronize with the primary nodes, keeping updated copies of the necessary state information. This ensures that, in case of a primary node failure, the standby node can take over the tasks with minimal disruption, maintaining consistency and reliability in the system. However, it is important to note that this approach incurs additional costs, particularly in terms of resource allocation and energy consumption. Maintaining standby nodes requires extra computational resources and power, which can impact the overall efficiency and sustainability of the system.

Mudassar et al. [15] works towards a solution that uses checkpointing and replication to manage the state of tasks. By periodically saving task states and replicating them across alternative edge nodes, this approach ensures that all replicas have the latest state information. Checkpointing captures the state of an application at specific intervals, and these checkpoints are then synchronized across multiple nodes. This method allows any replicated node to resume tasks from the last checkpoint in the event of a failure, reducing the impact on ongoing operations.

Redundancy also addresses communication and networking faults. Kulkarni et al. [22] propose “exactly-once” reliable message processing using platforms like Apache Storm/Trident and Apache Kafka. These platforms achieve message consistency and prevent duplication by employing mechanisms such as message deduplication, transaction logs, and consistent hashing. They ensure that the state is updated across all nodes involved in processing, maintaining message integrity even in the presence of failures.

Wang et al. [47] enhance messaging reliability by leveraging timing bounds to schedule message replication, ensuring reliable and timely delivery. This method ensures that messages are replicated within specified time intervals, maintaining the consistency of communication across nodes.

Khan et al. [63] explore metaverse-based wireless systems for proactive management and reliability. Their findings on virtual models and intelligent analytics can enhance redundancy and replication strategies in edge computing, improving system reliability and fault tolerance.

Within the metaverse, redundancy is particularly crucial for protecting user data and maintaining a seamless experience. Replicating critical data ensures its availability even during localized failures, minimizing data loss and preventing disruptions to user interactions.

4.4. Machine learning for predictive maintenance

Machine learning techniques analyze historical data and resource usage patterns to enable predictive maintenance and fault prevention. Tuli et al. [25] addresses resource exhaustion with preemptive task migration using PreGAN, a generative deep learning model that evaluates migration impacts before execution. While deep learning models like PreGAN are typically resource-intensive, their deployment in edge environments is carefully managed. Techniques such as model compression, quantization, and offloading complex computations to more capable nodes are employed to mitigate resource consumption.

Similarly, authors in [64] use machine learning to predict future system QoS. They implement models designed to balance the trade-off between accuracy and computational efficiency, ensuring they are suitable for edge devices. Algorithms such as support vector machines (SVM) and decision trees are commonly used for their efficiency and effectiveness in predictive maintenance.

Authors in [18,26] tackle the challenge of service disruptions and communication failures by using various machine learning models for fault prediction to preemptively initiate failover operations before failures occur. These models often use algorithms like random forests and gradient boosting, which provide robust predictions while being computationally manageable on edge devices.

Taking it a step further, Siyadatzadeh et al. [49] uses reinforcement learning in a novel primary-backup task assignment strategy named ReLIEF. This strategy employs Q-learning, a reinforcement learning algorithm that adapts the complexity of the learning model based on the current resource availability, ensuring that the reinforcement learning process remains efficient and does not consume excessive resources.

Early detection of potential issues like resource exhaustion or communication failures allows for proactive measures such as task migration or failover initiation, ensuring a smooth user experience in the metaverse. However, the effectiveness of this technique depends on the quality and comprehensiveness of the historical data used to train the models. Additionally, implementing resource-aware machine learning models is crucial for maintaining a balance between prediction accuracy and resource consumption, ensuring that the benefits of predictive maintenance outweigh the costs.

4.5. Consensus algorithms for security

Consensus algorithms are vital for maintaining integrity and security in distributed systems, particularly against Byzantine faults, ensuring all non-faulty nodes agree on a common state even when some nodes act maliciously.

Gao et al. [54] propose Improved Byzantine Consensus Mechanism based on K-medoids Clustering (FIBFT) to address scalability and performance issues in secure MEC applications. It clusters nodes based on performance and uses speculative Byzantine consensus within each cluster to reduce communication overhead and delay. Similarly, authors in [56] present Reja, a permissioned blockchain for secure storage, and ChiosEdge, an intrusion-tolerant system, to ensure data consistency and availability even with malicious devices.

To prevent misleading updates, Du et al. [58] introduce CRACAU, which identifies and excludes false updates from compromised devices during model training, maintaining the integrity of collaborative learning. Authors in [59] propose BytoChain, using verifiers to execute computationally expensive model verification tasks in parallel, and a Byzantine-resistant consensus mechanism based on Proof-of-Accuracy (PoA) to ensure only verified models contribute to the global model.

Consensus algorithms are crucial for metaverse security, ensuring all functioning nodes agree on a shared reality despite malicious actors. This is vital for data integrity and preventing inconsistencies within the virtual environment. The complexity of Byzantine fault tolerance is justified by the critical need for data integrity and user trust within the metaverse.

While these techniques are valuable for managing failures in edge environments, a comprehensive fault management strategy is necessary for the metaverse to thrive. The following section analyzes various fault management techniques through the lens of their applicability to the metaverse’s unique challenges.

Table 3
Summary of fault management techniques.

Author	Failure type	Category	Remediation strategy	Suitability for metaverse
Jia et al. [29]	Node failures for stateful applications	Proactive Monitoring	Monitors microservices using causal logging and distributed checkpointing for state reconstruction.	Ensures data consistency and restoration but might introduce overhead affecting real-time performance in metaverse applications.
Tuli et al. [25]	Node failure and resource exhaustion	Machine Learning for Predictive Maintenance	PreGAN predict task migration and use co-simulation to evaluate impacts before migration.	AI-powered migration benefits metaverse but complex models may add overhead. Further research needed.
Lei et al. [46]	Communication Faults and Service Invocation Failures	Task Scheduling and Workload Migration	Heuristic-based service binding algorithm and cache-enabled edge nodes for fault tolerance.	Improves fault tolerance but relies on historical data, requiring more research for metaverse integration.
Ray et al. [28]	Node failure and MEC server failures	Proactive Monitoring for Resource Optimization	Uses formal methods and heuristics to find optimal servers for container re-deployment.	Enhances fault tolerance for critical applications but relies on complete system information and complex computations.
Tuli et al. [64]	Broker node failures	Machine Learning for Predictive Maintenance	CAROL predicts QoS and fine-tunes models, reducing computational overhead.	Promises resilience and optimizes resource allocation in dynamic environments.
Liu et al. [45]	High task queue lengths, unreliable task execution	Proactive Monitoring for Resource Optimization	Uses extreme value theory and a two-timescale framework for resource optimization.	Ensures smooth user experiences but needs adaptation for metaverse-specific conditions.
Jing et al. [55]	Byzantine Faults	Proactive Monitoring	Proposes BFT consensus for edge computing, considering realistic failure models.	BFT consensus ensures reliable agreement in collaborative tasks; may need adaptation for metaverse.
Gao et al. [54]	Byzantine Faults	Consensus Algorithms for Security	FIBFT clusters nodes and uses speculative Byzantine consensus to reduce overhead.	Potential for metaverse by enhancing transaction efficiency and security with user/application-based clustering.
Wu et al. [56]	Byzantine Faults	Consensus Algorithms for Security	Reja uses permissioned blockchain and ChiosEdge for secure storage and data consistency.	Aligns with metaverse's need for user data ownership and privacy.
Mudassar et al. [15]	Node Failures	Redundancy and Replication	Uses checkpointing and replication to save task states and duplicate on alternative nodes.	Enhances reliability, but suitability depends on integration with real-time updates in metaverse.
Zhang et al. [57]	Byzantine Faults and Communication Failure	Consensus Algorithms for Security	CBFL addresses communication overhead and Byzantine attacks using local model updates.	Techniques for secure data aggregation could inspire metaverse data management.
Du et al. [58]	Byzantine Faults	Consensus Algorithms for Security	CRACAU identifies and excludes misleading updates from compromised devices during training.	Offers inspiration for mitigating malicious actors in collaborative virtual environments.
Li et al. [59]	Byzantine Faults	Consensus Algorithms for Security	BytoChain uses verifiers and PoA-based consensus to ensure only verified models contribute.	Techniques for achieving consensus can secure decentralized metaverse applications.
Kulkarni et al. [22]	Communication faults and Network Fluctuation	Redundancy and Replication	Evaluates overheads of reliable message processing in streaming platforms.	High responsiveness and reliability are crucial for metaverse, but evaluated reliability guarantees may hinder performance.
Li et al. [33]	Node Failures	Redundancy and Replication	M-MNFT selects redundant edge base stations based on proximity for task reliability.	Improves reliability but may not address rapid and unpredictable changes in metaverse.
Huang et al. [18]	Node Failures and Service Disruptions	Machine Learning for Predictive Maintenance	Uses machine learning for predictive failover operations.	Enhances reliability but may not directly translate to metaverse's dynamic and real-time requirements.
Sowmya et al. [19]	Edge server failures and Node failures	Machine Learning for Predictive Maintenance	Uses machine learning to detect potential failures and reroute traffic to backup servers.	Emphasizes availability and resilience, crucial for edge computing environments.
Cheng et al. [20]	Node Failures	Task Scheduling and Workload Migration	Two-stage adaptive robust optimization for resilient service placement and workload allocation.	Efficient resource management and resilience support real-time interactions in metaverse.
Taka et al. [24]	Node Failures	Task Scheduling and Workload Migration	Uses ILP and algorithms to minimize the worst-case penalty for user assignment and service placement.	Ensures reliable, low-latency service delivery even in the event of a BS failure, crucial for metaverse.
Wang et al. [47]	Communication Faults and Service Invocation Failures	Redundancy and Replication	FRAME schedules message delivery and replication to ensure reliability and timely delivery.	Ensures message consistency and reduces latency penalties during failures, enhancing real-time interactions.
Liu et al. [44]	Node Failures	Task Scheduling and Workload Migration	Enhances resilience through peer-collaborated service migration.	Ensures consistent performance and availability, crucial for immersive and interactive metaverse experiences.
Hou et al. [53]	Communication Faults and Service Invocation Failures	Task Scheduling and Workload Migration	LR-UMEN optimizes communications, computing, and caching resources.	Proactive resource management and failure anticipation enhance user experience in metaverse.

(*continued on next page*)

Table 3 (*continued*).

Author	Failure type	Category	Remediation strategy	Suitability for metaverse
Samir et al. [65]	Node Failures and Hardware Malfunctions	Machine Learning for Predictive Maintenance	Uses HMM for anomaly detection and prediction in edge cloud environments.	Helps maintain stable, low-latency services, but effectiveness depends on model scalability in metaverse.
Mohamed et al. [34]	Node Failures	Redundancy and Replication	Detects failures and ensures continuous operation in fog computing environments.	Mitigates disruptions in latency-sensitive applications, but effectiveness depends on scalability and real-time responsiveness.
Siyadatzadeh et al. [49]	Communication Faults and Service Invocation Failures	Machine Learning for Predictive Maintenance	ReLIEF uses RL for primary-backup task assignment in fog computing.	Ensures reliable execution of real-time tasks, adapting to changing conditions in fog network.
Martinez et al. [27]	Node failures	Redundancy and Replication	Uses standby fog nodes activated upon primary node failure, solved with a column generation approach.	Applicable for ensuring service continuity but might need adaptation for low-latency metaverse scenarios.
Soualhia et al. [66]	Node failures, network fluctuations	Machine Learning for Predictive Maintenance	Combines ML models for real-time fault detection and prediction using historical data.	Ensures service continuity but might need adaptation prioritizing low-latency over prediction accuracy.
Wang et al. [60]	Resource exhaustion	Proactive Monitoring for Resource Optimization	Combines client-side workload reduction and server-side resource allocation based on needs.	Ensures smooth user experiences but needs adaptation for metaverse-specific conditions.

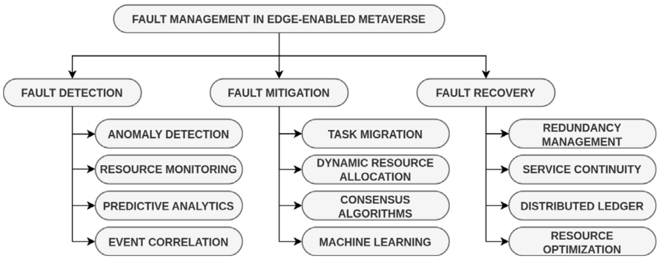


Fig. 4. Taxonomy of fault management approaches in edge-enabled metaverse applications.

5. Analysis of fault management techniques for metaverse applications

The metaverse presents a unique set of challenges for fault management. Real-time requirements, resource constraints and an expansive attack surface require robust solutions. This analysis, summarized in Table 3, critically examines various fault management techniques through the lens of their applicability to these metaverse requirements.

To systematically analyze these techniques, we propose a taxonomy of fault management approaches specifically designed for edge-enabled metaverse applications. This taxonomy categorizes the fault management strategies into three primary domains: Fault Detection, Fault Mitigation, and Fault Recovery. The flowchart in Fig. 4 illustrates this taxonomy, providing a clear framework for understanding the various techniques discussed in this section.

5.1. Ultra low latency & real-time requirements

Several techniques offer promise for identifying faults and managing system health, but adaptations are necessary for the metaverse's real-time demands. Techniques like causal logging and distributed checkpointing for state reconstruction [29] can ensure data consistency but might incur overhead impacting performance. These techniques are beneficial at the infrastructure layer, ensuring data integrity within the metaverse's distributed systems.

Proactive monitoring using formal methods and heuristics [28] for server re-deployment enhances fault tolerance but relies on complex computations potentially introducing latency. This method is useful for the content delivery networks (CDN) in edge environment, where anticipating and addressing faults can maintain the continuous availability and performance of metaverse services.

Machine Learning (ML) for predictive maintenance shows promise with techniques like CAROL [64] that predict Quality of Service (QoS) for resource optimization. This can be utilized in network infrastructure, ensuring uninterrupted data flow within the metaverse by preventing congestion.

Similarly, the approach in [66] combining ML models for real-time fault detection and prediction can ensure service continuity in user-facing applications. By prioritizing low-latency, these ML models maintain the responsiveness of interactive applications in metaverse.

Additionally, managing dynamic workloads is critical for optimizing resource utilization and performance across metaverse. Techniques like PreGAN for task migration prediction [25] and reinforcement learning for backup task assignment [49] offer adaptable solutions for fluctuating demands.

5.2. Resource constraints

The resource-constrained nature of edge devices necessitates lightweight and efficient solutions. Localized and efficient approaches like heuristic-based service binding algorithms with cache-enabled edge nodes [46] and client-side workload reduction with server-side resource allocation [60] are more suitable for the metaverse. These methods can help optimize resource use on edge devices, enhancing real-time interactions and computations within the metaverse's virtual environments and user interfaces.

Redundancy and replication techniques like standby fog nodes [27] and checkpointing with task replication [15] can enhance reliability. These techniques are vital for edge computing infrastructure, ensuring continuous operation and data integrity across the metaverse's distributed systems. However, their suitability depends on integration with real-time updates and scalability in the metaverse.

Similarly, offloading decision algorithms for optimized execution time, energy consumption, and reliability [44], along with M-MNFT for selecting redundant edge base stations [33], are suitable for network management and edge computing components. These approaches help balance load and improve reliability and energy efficiency, which is critical for maintaining a seamless and responsive metaverse experience.

5.3. Security threats in distributed systems

Byzantine Fault Tolerance (BFT) protocols are crucial for ensuring reliable agreement in collaborative tasks within the metaverse. Techniques like Improved Byzantine Consensus Mechanism based on K-medoids Clustering (FIBFT) [54] and Communication-Efficient and

Byzantine-Robust Federated Learning (CBFL) with local model updates [57] aim to reduce communication overhead and secure data aggregation in edge servers and distributed computing infrastructures.

Permissioned blockchain solutions with secure storage [56] manage user data ownership and transaction integrity in decentralized applications, safeguarding sensitive information across virtual environments. Techniques achieving consensus in decentralized applications, such as BytoChain [59], provide reliable data exchange and transaction validation, enhancing security in user interactions within the metaverse.

6. Case studies

To elucidate the practical applications of the fault management techniques explored, we present a series of case studies within the metaverse context. These case studies are designed to illustrate the effectiveness of specific solutions in addressing key challenges such as network latency, virtual asset corruption, server overload, and security breaches.

6.1. Network latency and packet loss

In the metaverse environment, network latency and packet loss can severely disrupt real-time interactions. To address these challenges, techniques such as heuristic-based service binding algorithms and cache-enabled edge nodes, as proposed by Lei et al. [46], offer robust solutions. These methods improve fault tolerance by optimizing service binding and reducing latency through local caching at the edge. Additionally, proactive monitoring solutions, such as those utilizing formal methods and heuristics [28], can enhance fault tolerance by identifying and mitigating potential network issues before they impact the metaverse experience. By combining these techniques, it is possible to ensure smoother and more reliable network performance, crucial for maintaining immersive interactions in the metaverse.

6.2. Virtual asset corruption

Virtual asset corruption in the metaverse can compromise user experience and data integrity. To combat this, causal logging and distributed checkpointing, as explored by Jia et al. [29], provide an effective strategy. This approach ensures data consistency by reconstructing the state of virtual assets from logs and checkpoints, thus mitigating the impact of corruption. Additionally, redundancy and replication techniques, such as those using standby fog nodes [27], can enhance reliability by providing backup instances of virtual assets. Together, these solutions help maintain the integrity and continuity of virtual assets, ensuring a stable and engaging metaverse environment.

6.3. Server overload

Resource overload can lead to significant performance degradation in metaverse applications. To address this issue, proactive monitoring for resource optimization, as described by Wang et al. [60], and machine learning techniques for predictive maintenance, such as CAROL [64], can be employed. Proactive monitoring involves optimizing server resource allocation and reducing client-side workloads to manage server demands effectively. Meanwhile, predictive maintenance models can anticipate potential overloads and facilitate timely resource adjustments. By implementing these strategies, it is possible to enhance server performance and prevent overload situations, ensuring consistent and responsive service delivery within the metaverse.

6.4. Security breach

Security breaches in the metaverse can undermine user trust and data integrity. Techniques like Byzantine Fault Tolerance (BFT) mechanisms and permissioned blockchain solutions offer robust security measures. For instance, the Improved Byzantine Consensus Mechanism (FIBFT) [54] and the Communication-Efficient and Byzantine-Robust Federated Learning (CBFL) [57] provide secure data aggregation and consensus methods that enhance resilience against malicious attacks. Additionally, permissioned blockchain solutions, such as those employing secure storage [56], ensure the integrity and confidentiality of user data. These solutions collectively strengthen the security framework of the metaverse, safeguarding user interactions and protecting against potential breaches.

7. Discussion & future research directions

The preceding sections have discussed the fault tolerance strategies in edge computing and how they can help in addressing challenges in metaverse. While these techniques show promise for ensuring continuous service and user satisfaction, significant gaps persist.

This detailed analysis reveals three major challenges in enabling metaverse: efficient fault prediction on constrained devices, data bottlenecks during task migration and secure data sharing.

The metaverse's distributed nature, relying on dispersed edge devices and dynamic user interactions, requires further research to tackle these issues. This section explores these challenges and suggests future research directions to strengthen failure remediation in the metaverse.

7.1. Efficient fault prediction on edge devices

The potential of machine learning for fault prediction within the edge-enabled metaverse has been previously discussed. Future research can delve deeper into:

- *Resource Aware ML Models:* Development of lightweight machine learning models designed for deployment on resource-constrained edge devices can significantly reduce computational complexity while maintaining prediction accuracy. These models can ensure responsive user experiences by eliminating high bandwidth transfers to cloud servers for processing.
- *Federated Learning for Fault Prediction:* Exploration of federated learning, where models are trained collaboratively across edge devices without compromising user data privacy. This leverages the collective resources of edge devices for fault prediction accuracy while safeguarding privacy, resulting in better security of metaverse users.
- *Deterministic Failure Handling:* To address deterministic failures, future work could explore techniques that steer execution away from error-prone paths. Notable research in this area includes Crystalball [67], which predicts and prevents inconsistencies in distributed systems by analyzing historical execution patterns and detecting potential inconsistencies before they impact system performance. Similarly, Bouncer [68] secures software by blocking bad input, preventing faults from arising due to malicious or erroneous inputs. Tardis [69] provides a fault-tolerant design for network control planes by using temporal redundancy and error-correcting codes to ensure that network control messages are resilient to failures. These approaches can be adapted to edge environments to enhance the robustness of fault prediction and prevention strategies.

7.2. Data bottlenecks during migration

A critical challenge highlighted in this work concerns the impact of data locality on task migration strategies. While workload migration enhances resource utilization, it might not be equally beneficial for tasks with high data dependency. Future research efforts could target:

- *LLM-powered Contextual Summarization:* Development of Large Language Models (LLMs) capable of generating concise, high-fidelity summaries of user-specific data would allow tasks to leverage migrated resources while retaining access to crucial insights from user data residing on the original node. This would empower metaverse personalization engines to provide features such as user preferences and behaviors across metaverse applications.
- *Predictive Data Pre-fetching:* Investigation of machine learning techniques that anticipate resource demands and pre-fetch necessary data to the target node before task migration occurs. By pre-fetching data, performance impact of data transfer during migration can be minimized, leading to seamless experiences within the metaverse.
- *Incorporating User Feedback:* Integrating user feedback into failure detection can enhance the accuracy and relevance of fault management strategies. Notable work in this area includes research by Venkataraman et al. [70,71], which explores methods for utilizing user-reported issues to improve network reliability. On the other hand, CableMon, proposed by Hu et al. [72], focuses on proactive maintenance by analyzing user feedback to predict and address potential network issues before they impact users. Incorporating such feedback mechanisms into edge computing environments can provide valuable insights into real-world failures and help refine fault prediction models.

7.3. Data security and integrity

Redundancy and replication are crucial for ensuring data availability and integrity within the metaverse. However, traditional replication strategies might not be efficient for geographically distributed edge deployments. Future research can explore:

- *Blockchain-based Secure Data Sharding:* Blockchain can leverage its tamper-proof nature to ensure data consistency across geographically distributed edge nodes. By dividing data into shards and storing them across various edge devices, blockchain provides decentralized storage and an immutable record of data locations. This transparency prevents unauthorized modifications and enhances security. Consensus mechanisms like Proof of Stake (PoS) or Practical Byzantine Fault Tolerance (PBFT) ensure accurate and consistent replication, while smart contracts automate data retrieval. This approach mitigates the risks associated with traditional replication strategies and offers a robust solution for managing vast amounts of data across edge nodes. This is particularly useful for real-time applications in the metaverse that require quick access to distributed data. Moreover, blockchain-based sharding can be scaled horizontally by adding more nodes to the network, ensuring that data storage and retrieval processes remain efficient and reliable as the metaverse grows.
- *Dynamic Data Replication Protocols:* Development of self-healing replication protocols that dynamically adjust replication levels based on real-time network conditions and fault occurrences. This research can benefit the metaverse's storage layer by ensuring reliable storage of vast amounts of data across the distributed edge network.

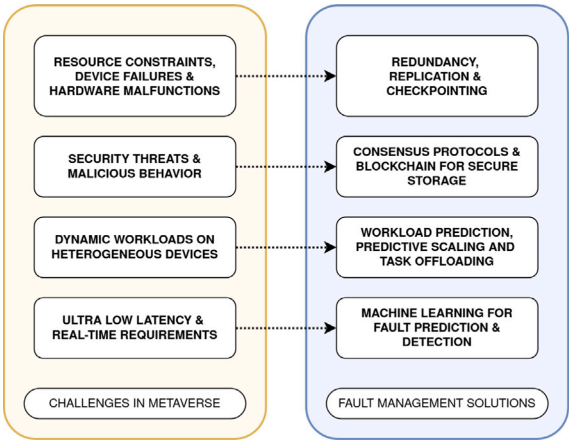

RESOURCE CONSTRAINTS, DEVICE FAILURES & HARDWARE MALFUNCTIONS
REDUNDANCY, REPLICATION & CHECKPOINTING
SECURITY THREATS & MALICIOUS BEHAVIOR
CONSENSUS PROTOCOLS & BLOCKCHAIN FOR SECURE STORAGE
DYNAMIC WORKLOADS ON HETEROGENEOUS DEVICES
WORKLOAD PREDICTION, PREDICTIVE SCALING AND TASK OFFLOADING
ULTRA LOW LATENCY & REAL-TIME REQUIREMENTS
MACHINE LEARNING FOR FAULT PREDICTION & DETECTION
CHALLENGES IN METAVERSE
FAULT MANAGEMENT SOLUTIONS

Fig. 5. Analysis of metaverse challenges.

8. Conclusion

While the current landscape of fault management techniques provides valuable tools for building reliable distributed metaverses, significant research gaps remain. In this paper, we explored the common types of faults and failures in edge computing, their existing solutions, and their applicability to challenges faced in edge-enabled metaverses. A visualization of this analysis, presented in Fig. 5, offers a comprehensive comparison.

Existing solutions, despite offering a foundation for redundancy, workload management, and proactive monitoring, often lack the adaptability and real-time responsiveness necessary for the metaverse's evolving and dynamic nature. Furthermore, security considerations within the metaverse, especially regarding Byzantine faults and user-generated content, necessitate further exploration of consensus algorithms specifically tailored to these challenges. These limitations underscore the need for a paradigm shift towards holistic fault management frameworks designed for the unique requirements of the metaverse.

CRediT authorship contribution statement

Shahzaib Shaikh: Methodology, Investigation, Formal analysis. **Manar Jammal:** Writing – review & editing, Supervision, Resources, Project administration.

Declaration of competing interest

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

Data availability

No data was used for the research described in the article.

References

[1] C. Zhang, X. Si, X. Zhu, Y. Zhang, A survey on the security of the metaverse, in: IEEE International Conference on Metaverse Computing, Networking and Applications (MetaCom), 2023, pp. 428–432.
[2] D.H. Nam, A comparative study of mobile cloud computing, mobile edge computing, and mobile edge cloud computing, in: Congress in Computer Science, Computer Engineering, & Applied Computing, CSCE, 2023, pp. 1219–1224.
[3] Thegamingwatcher.com, Xbox Cloud outage disrupts gaming experience for users, in: The Gaming Watcher, thegamingwatcher.com, 2024.
[4] Thegamingwatcher.com, Steam servers down: Gamers report outage and server connection issues, in: The Gaming Watcher, thegamingwatcher.com, 2024.

[5] M. Satyanarayanan, P. Bahl, R. Caceres, N. Davies, The case for VM-based cloudlets in mobile computing, IEEE Pervasive Comput. 8 (4) (2009) 14–23.

[6] P. Cruz, N. Achir, A.C. Viana, On the edge of the deployment: A survey on multi-access edge computing, ACM Comput. Surv. 55 (5) (2022).

[7] F. Bonomi, R. Milito, J. Zhu, S. Addepalli, Fog computing and its role in the internet of things, in: Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing, MCC '12, Association for Computing Machinery, New York, NY, USA, 2012, pp. 13–16.

[8] W. Kong, X. Li, L. Hou, J. Yuan, Y. Gao, S. Yu, A reliable and efficient task offloading strategy based on multifeedback trust mechanism for IoT edge computing, IEEE Internet Things J. 9 (15) (2022) 13927–13941, http://dx.doi.org/10.1109/JIOT.2022.3143572.

[9] S. Suryavansh, C. Bothra, M. Chiang, C. Peng, S. Bagchi, Tango of edge and cloud execution for reliability, in: Proceedings of the 4th Workshop on Middleware for Edge Clouds & Cloudlets, MECC '19, Association for Computing Machinery, New York, NY, USA, 2019, pp. 10–15.

[10] K. Ergun, R. Ayoub, P. Mercati, T.S. Rosing, Dynamic reliability management of multigateway IoT edge computing systems, IEEE Internet Things J. 10 (5) (2023) 3864–3889.

[11] W. Wang, D. Wu, S. Das, A. Rahbar, A. Chen, T.S.E. Ng, RDC: Energy-efficient data center network congestion relief with topological reconfigurability at the edge, in: 19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22), USENIX Association, Renton, WA, 2022, pp. 1267–1288.

[12] V. Pourahmadi, H.A. Alameddine, M.A. Salahuddin, R. Boutaba, Spotting anomalies at the edge: Outlier exposure-based cross-silo federated learning for DDoS detection, IEEE Trans. Dependable Secure Comput. 20 (5) (2023) 4002–4015, http://dx.doi.org/10.1109/TDSC.2022.3224896.

[13] A. Sathiaseelan, M. Selimi, C. Molina, A. Lertsinsrubtavee, L. Navarro, F. Freitag, F. Ramos, R. Baig, Towards decentralised resilient community clouds, in: Proceedings of the 2nd Workshop on Middleware for Edge Clouds & Cloudlets, MECC '17, Association for Computing Machinery, New York, NY, USA, 2017.

[14] C. Correia, Safeguarding data consistency at the edge, in: 50th Annual IEEE-IFIP International Conference on Dependable Systems and Networks-Supplemental Volume (DSN-S), 2020, pp. 65–66.

[15] M. Mudassar, Y. Zhai, L. Lejian, Adaptive fault-tolerant strategy for latency-aware IoT application executing in edge computing environment, IEEE Internet Things J. 9 (15) (2022) 13250–13262.

[16] X. Liu, J. Jiang, L. Li, Computation offloading and task scheduling with fault-tolerance for minimizing redundancy in edge computing, in: 2021 IEEE International Symposium on Software Reliability Engineering Workshops, ISSREW, 2021, pp. 198–209.

[17] Y. Li, X. Sun, Y. Xia, P. Chen, Y. Li, Q. Peng, M-MNFT: A novel modified (m, n)-fault tolerance approach for service migration in vehicular edge computing, in: IEEE International Conference on Software Services Engineering, SSE, 2023, pp. 170–179.

[18] H. Huang, S. Guo, Proactive failure recovery for NFV in distributed edge computing, IEEE Commun. Mag. 57 (5) (2019) 131–137.

[19] S. R, N. M, P. M, Enhancing edge node resilience through SDN-driven proactive failure management, in: IEEE International Conference for Women in Innovation, Technology and Entrepreneurship, ICWITE, 2024, pp. 15–20.

[20] J. Cheng, D.T. Nguyen, V.K. Bhargava, Resilient edge service placement under demand and node failure uncertainties, IEEE Trans. Netw. Serv. Manag. 21 (1) (2024) 558–573.

[21] J. Zilic, A. Aral, I. Brandic, EFPO: Energy efficient and failure predictive edge offloading, in: Proceedings of the 12th IEEE/ACM International Conference on Utility and Cloud Computing, UCC '19, Association for Computing Machinery, New York, NY, USA, 2019, pp. 165–175.

[22] S.G. Kulkarni, G. Liu, K.K. Ramakrishnan, T. Wood, Living on the edge: Serverless computing and the cost of failure resiliency, in: 2019 IEEE International Symposium on Local and Metropolitan Area Networks, LANMAN, 2019, pp. 1–6.

[23] C. Tang, H. Wu, C. Zhu, Satisfaction optimization in failure-aware vehicular edge computing, in: IEEE Global Communications Conference, 2022, pp. 5783–5788.

[24] H. Taka, F. He, E. Oki, Service placement and user assignment in multi-access edge computing with base-station failure, in: 2022 IEEE/ACM 30th International Symposium on Quality of Service (IWQoS), 2022, pp. 1–10.

[25] S. Tuli, G. Casale, N.R. Jennings, PreGAN: Preemptive migration prediction network for proactive fault-tolerant edge computing, in: IEEE Conference on Computer Communications, 2022, pp. 670–679.

[26] M. Soualhia, C. Fu, F. Khomh, Infrastructure fault detection and prediction in edge cloud environments, in: Proceedings of the 4th ACM/IEEE Symposium on Edge Computing, SEC '19, Association for Computing Machinery, New York, NY, USA, 2019, pp. 222–235.

[27] I. Martinez, A.S. Hafid, M. Gendreau, Robust and fault-tolerant fog design and dimensioning for reliable operation, IEEE Internet Things J. 9 (19) (2022) 18280–18292.

[28] K. Ray, A. Banerjee, Prioritized fault recovery strategies for multi-access edge computing using probabilistic model checking, IEEE Trans. Dependable Secure Comput. 20 (1) (2023) 797–812.

[29] Y. Jia, T. Wang, T. Qiu, X. Zhang, R. Wang, T. Wo, Fault tolerance of stateful microservices for industrial edge scenarios, in: IEEE International Conference on Joint Cloud Computing, JCC, IEEE Computer Society, Los Alamitos, CA, USA, 2023, pp. 50–56.

[30] X. Chen, G. Xu, X. Xu, H. Jiang, Z. Tian, T. Ma, Multicenter hierarchical federated learning with fault-tolerance mechanisms for resilient edge computing networks, IEEE Trans. Neural Netw. Learn. Syst. (2024) 1–15.

[31] A. Kouloumpris, M.K. Michael, T. Theocharides, Reliability-aware task allocation latency optimization in edge computing, in: IEEE 25th International Symposium on on-Line Testing and Robust System Design, IOLTS, 2019, pp. 200–203.

[32] P. Zhao, G. Dán, A benders decomposition approach for resilient placement of virtual process control functions in mobile edge clouds, IEEE Trans. Netw. Serv. Manag. 15 (4) (2018) 1460–1472.

[33] B. Li, Q. He, G. Cui, X. Xia, F. Chen, H. Jin, Y. Yang, READ: Robustness-oriented edge application deployment in edge computing environment, IEEE Trans. Serv. Comput. 15 (3) (2022) 1746–1759.

[34] N. Mohamed, J. Al-Jaroodi, I. Jawhar, Towards fault tolerant fog computing for IoT-based smart city applications, in: IEEE 9th Annual Computing and Communication Workshop and Conference, CCWC, 2019, pp. 0752–0757.

[35] P. Zhang, Y. Chen, M. Zhou, G. Xu, W. Huang, Y. Al-Turki, A. Abusorrah, A fault-tolerant model for performance optimization of a fog computing system, IEEE Internet Things J. 9 (3) (2022) 1725–1736.

[36] L. Dong, Q. Ni, W. Wu, C. Huang, T. Znati, D.Z. Du, A proactive reliable mechanism-based vehicular fog computing network, IEEE Internet Things J. 7 (12) (2020) 11895–11907.

[37] M.M. Tajiki, M. Shojafar, B. Akbari, S. Salsano, M. Conti, M. Singhal, Joint failure recovery, fault prevention, and energy-efficient resource management for real-time SFC in fog-supported SDN, Comput. Netw. 162 (C) (2019).

[38] N. Vance, M.T. Rashid, D. Zhang, D. Wang, Towards reliability in online high-churn edge computing: A deviceless pipelining approach, in: IEEE International Conference on Smart Computing, SMARTCOMP, 2019, pp. 301–308.

[39] J. Zhao, M. Dai, Y. Xia, Y. Ma, M. He, K. Peng, J. Li, F. Li, X. Fu, FRSM: A novel fault-tolerant approach for redundant-path-enabled service migration in mobile edge computing, in: 29th International Conference, Held As Part of the Services Conference Federation, SCF 2022, Honolulu, HI, USA, December 10–14, 2022, Proceedings, Springer-Verlag, Berlin, Heidelberg, 2022, pp. 1–12.

[40] M. Whaiduzzaman, A. Barros, A.R. Shovon, M.R. Hossain, C. Fidge, A resilient fog-IoT framework for seamless microservice execution, in: IEEE International Conference on Services Computing, SCC, 2021, pp. 213–221.

[41] Y. Ramzanpoor, M. Hosseini Shirvani, M. Golsorkhtabaramiri, Multi-objective fault-tolerant optimization algorithm for deployment of IoT applications on fog computing infrastructure, Complex Intell. Syst. 8 (1) (2022) 361–392.

[42] S. Ghanavati, J. Abawajy, D. Izadi, Automata-based dynamic fault tolerant task scheduling approach in fog computing, IEEE Trans. Emerg. Top. Comput. 10 (1) (2022) 488–499.

[43] L. Cai, X. Wei, C. Xing, X. Zou, G. Zhang, X. Wang, Failure-resilient DAG task scheduling in edge computing, Comput. Netw. 198 (2021) 108361.

[44] K. Liu, N. Manangi Ravindrarao, A. Gurudutt, T. Kamaal, C. Divakara, P. Prabhakaran, Y. Chen, Software-defined edge cloud framework for resilient multitenant applications, Wirel. Commun. Mob. Comput. 2019 (2019).

[45] C.-F. Liu, M. Bennis, M. Debbah, H.V. Poor, Dynamic task offloading and resource allocation for ultra-reliable low-latency edge computing, IEEE Trans. Commun. 67 (6) (2019) 4132–4150.

[46] C. Lei, H. Dai, A heuristic services binding algorithm to improve fault-tolerance in microservice based edge computing architecture, in: IEEE World Congress on Services, SERVICES, 2020, pp. 83–88.

[47] C. Wang, C. Gill, C. Lu, FRAME: Fault tolerant and real-time messaging for edge computing, in: IEEE 39th International Conference on Distributed Computing Systems, ICDCS, 2019, pp. 976–985.

[48] E.E. Haber, H.A. Alameddine, C. Assi, S. Sharafeddine, A reliability-aware computation offloading solution via UAV-mounted cloudlets, in: IEEE 8th International Conference on Cloud Networking (CloudNet), 2019, pp. 1–6.

[49] R. Siyadatzadeh, F. Mehrafrooz, M. Ansari, B. Safaei, M. Shafique, J. Henkel, A. Ejlali, Relief: A reinforcement-learning-based real-time task assignment strategy in emerging fault-tolerant fog computing, IEEE Internet Things J. 10 (12) (2023) 10752–10763.

[50] M. Chen, S. Guo, K. Liu, X. Liao, B. Xiao, Robust computation offloading and resource scheduling in cloudlet-based mobile cloud computing, IEEE Trans. Mob. Comput. 20 (5) (2021) 2025–2040.

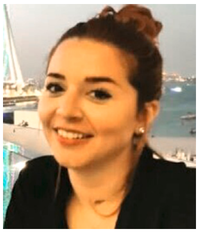
[51] A. Lakhan, X. Li, Transient fault aware application partitioning computational offloading algorithm in microservices based mobile cloudlet networks, Computing 102 (1) (2020) 105–139.

[52] A. Aral, I. Brandić, Learning spatiotemporal failure dependencies for resilient edge computing services, IEEE Trans. Parallel Distrib. Syst. 32 (7) (2021) 1578–1590.

[53] X. Hou, Z. Ren, J. Wang, S. Zheng, H. Zhang, Latency and reliability oriented collaborative optimization for multi-UAV aided mobile edge computing system, in: IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2020, pp. 150–156.

[54] N. Gao, R. Huo, S. Wang, T. Huang, FIBFT: An improved Byzantine consensus mechanism for edge computing, in: IEEE Wireless Communications and Networking Conference, WCNC, 2023, pp. 1–6.
[55] G. Jing, Y. Zou, D. Yu, C. Luo, X. Cheng, Efficient fault-tolerant consensus for collaborative services in edge computing, IEEE Trans. Comput. 72 (8) (2023) 2139–2150.
[56] Y. Wu, J. Liao, P. Nguyen, W. Shi, Y. Yesha, Bring trust to edge: Secure and decentralized IoT framework with bft and permissioned blockchain, in: IEEE International Conference on Edge Computing and Communications, EDGE, 2022, pp. 104–113.
[57] Z. Zhang, L. Wu, D. He, J. Li, S. Cao, X. Wu, Communication-efficient and Byzantine-robust federated learning for mobile edge computing networks, IEEE Netw. 37 (4) (2023) 112–119.
[58] A. Du, Y. Shen, Q. Zhang, L. Tseng, M. Aloqaily, CRACAU: Byzantine machine learning meets industrial edge computing in industry 5.0, IEEE Trans. Ind. Inform. 18 (8) (2022) 5435–5445.
[59] Z. Li, H. Yu, T. Zhou, L. Luo, M. Fan, Z. Xu, G. Sun, Byzantine resistant secure blockchained federated learning at the edge, IEEE Netw. 35 (4) (2021) 295–301.
[60] J. Wang, Z. Feng, S. George, R. Iyengar, P. Pillai, M. Satyanarayanan, Towards scalable edge-native applications, in: Proceedings of the 4th ACM/IEEE Symposium on Edge Computing, SEC '19, Association for Computing Machinery, New York, NY, USA, 2019, pp. 152–165, http://dx.doi.org/10.1145/3318216.3363308.
[61] P. Thantharate, IntelligentMonitor: Empowering DevOps environments with advanced monitoring and observability, in: International Conference on Information Technology, ICIT, 2023, pp. 800–805.
[62] H. Qiu, K. Zhu, D. Niyato, Secure and flexible coded distributed matrix multiplication based on edge computing for industrial metaverse, IEEE Trans. Cloud Comput. (01) 1–15, 5555.
[63] L.U. Khan, M. Guizani, D. Niyato, A. Al-Fuqaha, M. Debbah, Metaverse for wireless systems: Architecture, advances, standardization, and open challenges, Internet Things 25 (2024) 101121.
[64] S. Tuli, G. Casale, N.R. Jennings, CAROL: Confidence-aware resilience model for edge federations, in: 52nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN, IEEE Computer Society, Los Alamitos, CA, USA, 2022, pp. 28–40.
[65] A. Samir, C. Pahl, Detecting and predicting anomalies for edge cluster environments using hidden Markov models, in: Fourth International Conference on Fog and Mobile Edge Computing, FMEC, 2019, pp. 21–28.
[66] M. Soualhia, C. Fu, F. Khomh, Infrastructure fault detection and prediction in edge cloud environments, in: Proceedings of the 4th ACM/IEEE Symposium on Edge Computing, SEC '19, Association for Computing Machinery, New York, NY, USA, 2019, pp. 222–235.
[67] M. Yabandeh, N. Knezevic, D. Kostic, V. Kuncak, CrystalBall: predicting and preventing inconsistencies in deployed distributed systems, in: Proceedings of the 6th USENIX Symposium on Networked Systems Design and Implementation, NSDI '09, USENIX Association, USA, 2009, pp. 229–244.
[68] M. Costa, M. Castro, L. Zhou, L. Zhang, M. Peinado, Bouncer: securing software by blocking bad input, in: Proceedings of Twenty-First ACM SIGOPS Symposium on Operating Systems Principles, SOSP '07, Association for Computing Machinery, New York, NY, USA, 2007, pp. 117–130.
[69] Z. Zhou, T.A. Benson, M. Canini, B. Chandrasekaran, Tardis: A fault-tolerant design for network control planes, in: Proceedings of the ACM SIGCOMM Symposium on SDN Research (SOSR), SOSR '21, Association for Computing Machinery, New York, NY, USA, 2021, pp. 108–121.
[70] S. Venkataraman, J. Wang, Towards identifying impacted users in cellular services, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD '19, Association for Computing Machinery, New York, NY, USA, 2019, pp. 3029–3039.
[71] S. Venkataraman, J. Wang, Assessing the impact of network events with user feedback, in: Proceedings of the 2018 Workshop on Network Meets AI & ML, NetAI '18, Association for Computing Machinery, New York, NY, USA, 2018, pp. 74–79.
[72] J. Hu, Z. Zhou, X. Yang, J. Malone, J.W. Williams, CableMon: improving the reliability of cable broadband networks via proactive network maintenance, in: Proceedings of the 17th Usenix Conference on Networked Systems Design and Implementation, NSDI '20, USENIX Association, USA, 2020, pp. 619–632.

Shahzaib Shaikh is currently pursuing a master's degree in Information Systems & Technology at York University. He holds a BS in Electrical Engineering from Habib University. His research primarily focuses on failure prediction and self-healing applications in edge and cloud computing systems, leveraging machine learning techniques to enhance system reliability and performance. With a strong background in both theoretical and practical aspects of information systems, Shahzaib aims to contribute to the advancement of predictive technologies in distributed computing environments.



Manar Jammal is an Associate Professor at the School of IT at York University. She holds a B.E.Sc. in Electrical and Computer Engineering from Lebanese University, an M.E.Sc. from the University of Technology of Compiegne, and a Ph.D. in Software Engineering from Western University. She has two invention disclosures with Ericsson for managing carrier-grade cloud applications. Her research focuses on AI governance, machine learning, data analytics, 5G, IoT, smart cities, and cloud computing. Dr. Jammal is a Review Editor for Frontiers in Communications and Networks and regularly reviews for IEEE and other journals and conferences.