Indonesian Journal of Computer Science
ISSN 2549-7286 (*online*)
Jln. Khatib Sulaiman Dalam No. 1, Padang, Indonesia
Website: ijcs.stmikindonesia.ac.id | E-mail: ijcs@stmikindonesia.ac.id

Proactive Fault Tolerance in Distributed Cloud Systems: A Review of Predictive and Preventive Techniques

Dathar Abas Hasan [1,2], Subhi R. M. Zeebaree [3]

dathar.hasan@dpu.edu.krd, subhi.rafeeq@dpu.edu.krd

[1] Information Technology Department, Technical College of Informatics, Akre University of Applied Sciences, Duhok, Iraq.

[2] Public Health Department, College of Health and Medical Technology – Shekhan, Duhok Polytechnic University, Duhok, Iraq.

[3] Energy Engineering Department, Technical College of Engineering, Duhok Polytechnic University, Duhok, Iraq.

Article Information

Submitted : 5 Mar 2024
Reviewed: 16 Mar 2024
Accepted : 1 Apr 2024

Keywords

Cloud Computing, Fault Tolerance, Proactive Techniques, Predictive Techniques, Cloud Availability.

Abstract

In a cloud computing environment, various hardware and software services are provided to the users across multiple servers and data centers. These servers are communicated to each other to allow greater scalability, flexibility, and reliability. Reliability is a vital factor in cloud computing that ensures that the requested services will be delivered to the users whenever they request them. However, different hardware or software faults may occur in cloud servers or data centers that prevent the users from receiving the service. Fault tolerance is defined as the ability of the system to provide services to the users even with the presence of faults or failures. In this review, we focused on some of the emerging fault tolerance techniques researchers have proposed to tackle the fault issues in cloud computing. We divided these techniques into two main categories: proactive and reactive techniques. Proactive techniques involve protecting the system defects by proposing certain procedures to prevent reaching the defective condition. Reactive techniques refer to the ability of the cloud system to recover the defective server or framework to continue working and providing the service.

A. Introduction

Recently, there has been an increasing demand for cloud computing services worldwide. These services may include data processing, online applications, virtual machines, and online storage. These services are provided by cloud agents with significant features, the users can access these services from remote locations with relatively low costs [1][2]. Cloud computing services offer a variety of services that can be requested at specific times and provided continuously without interruption. These services are provided through numerous servers and data centers located in many countries around the world to ensure the best service with the lowest response time and competitive prices [3][4]. Cloud service providers work continuously to achieve the highest level of reliability in their services. However, the large number of interconnected servers through the Internet may sometimes face various types of faults that can affect the quality of the provided service or cause service interruption for some users [5][6].

The reliability factor in cloud computing services can be directly affected by some malfunctions or deficiencies in the cloud infrastructure. This hurts the quality of service provided to users [7]. Therefore, it is essential to adopt an early detection system for faults and failures to repair and restore the malfunctioning parts to continue operating, this could prevent the expansion of the fault and ensure that the service is presented continuously with sufficient quality [8]. Fault tolerance in cloud computing tends to deal with unexpected issues that result from errors and failures. To enhance reliability and achieve resilience in cloud computing, defects must be properly analyzed and addressed [9]. In cloud computing systems, there are two types of fault tolerance: hardware fault tolerance and software fault tolerance [10]. Fault tolerance is crucial as cloud resources are provided for error detection, identification, recovery, and enhancing user performance efficiency. Therefore, fault tolerance has become a vital challenge for cloud computing systems. The Key benefits of fault tolerance modification include reduced costs, error recovery, and improved performance indicators [11].

Cloud vendors should pay attention to checking their cloud systems continuously for any failure or errors that may appear in any component or stage within the cloud environment. The focus should be on when, where, and how the failure may appear and how to prevent them [12]. However, recognizing the faults before a collapse is an important step, while recognizing the fault's nature is more important to take appropriate action and ensure that it will not appear again in the future [13]. Some faults can generate unavailable resources that cause the web applications and servers to fall apart The continuous changes in the cloud hardware and software force the current fault tolerance techniques to be updated to recognize conceptual differences between normal cloud variation and real-time failures [14].

In this review, we focused on the many research studies that have proposed promised fault tolerance techniques to handle errors and failures in cloud computing. This review is organized as follows: The second section presents the main concepts about distributed and cloud systems. The third section involves the theoretical background of fault tolerance in cloud computing. The fourth section includes a literature review of some of the recent studies and techniques. In the fifth section, we summarized and discussed the proposed techniques. Section Six includes some recommendations to improve the fault tolerance in cloud systems. Finally, section seven presents multiple conclusions from this study.

B. Main Concepts of Distributed and Cloud Systems

1. Distributed Systems

Distributed System is defined as multiple, independent, heterogeneous, and physically separated computer systems that are connected via a centralized network to share the files and resources for executing specified goals. The computer is referred to as a node in the distributed systems [15].

1.1 Benefits of The Distributed Systems

The main three benefits of the distributed systems can be summarized as follows:

- **Scalability:** As a task is executed by a computer separately, therefore, expanding the system by adding more nodes to increase the performance is an easy and cost-effective process [16].
- **Reliability:** The ability to continue executing the tasks through hundreds of nodes even with a failure in a single or multiple nodes is a crucial factor in the distribution system [17].
- **Performance:** The efficiency of the distributed systems is optimized as the workloads are distributed on multiple nodes to be executed separately [18].

1.2 Architectures of Distributed Systems

In general, distributed systems can be classified into three main architectures:

- **Client-Server:** Clients (nodes) receive data (tasks) from the server, execute these tasks, share the results, and finally store the results in the server [19].
- **Three-tier:** Dividing the system components into three separate layers. Each layer is responsible for performing specific tasks. It is very useful for security and scalability concepts [20].
- **Peer-to-peer:** Each node can be a server and client and sends or receives data to or from another node directly. There are uniform responsibilities among nodes in the system [20].

2. Cloud System

Cloud computing is a new computing style that accesses, processes, and stores data or programs over the Internet from remote sites. It can be done anywhere, anytime using any computer device [19].

2.1 Cloud service models

Cloud systems can present services in three models as follows:

- **Infrastructure as a Service (IaaS):** It enables the users to access the storage, networking, servers, and other computing resources via the cloud with a high level of control and flexibility [20].
- **Platform as a Service (PaaS):** This model presents its services to developers. It involves leasing a variety of cloud-based platforms to the users for the building and development of multi-purpose applications [21].
- **Software as a Service (SaaS):** Delivers various applications as a service over the Internet instead of installing on the user's computers. It provides cost-effective, flexible, and reliable applications to users [22].

2.2 Cloud Deployment Models

- **Private Cloud:** It is owned by a single organization, this organization manages, maintains, and operates the cloud to present services to the organization's users only [23].
- **Private Cloud:** It can be owned, managed, and operated by a business, academic, or government organization. The services are presented to the general public [24].
- **Community Cloud:** It is owned by several organizations; it is managed by the organizations or a third party. The services are presented to users of the specific community [24].
- **Hybrid Cloud:** It is a composition of many cloud infrastructures (private, community, or public). It enables data and application portability (e.g., cloud bursting for load balancing between clouds) [24].

C. Fault Tolerance in Cloud Computing

Fault tolerance is considered an important criterion to evaluate the performance level of the provided services in cloud computing. Any lack of cloud resources could lead to a vital issue in response time, throughput, and job execution [25]. Therefore, adopting a robust fault tolerance system is an essential step for error identification and overcoming. As the cloud computing architecture is designed based on a large number of interconnected nodes, hence, a crash of one node can affect the entire cloud network [26]. The ability to continue executing the requested tasks in the presence of internal defects is a crucial objective of the fault tolerances in the cloud system. Real-time applications require providing on-time services with a high level of availability and reliability to the users [27].

Three important concepts are usually used in computer and cloud systems which are fault, error, and failure. The term "fault" in the computer refers to the system's inability to perform the required functions due to unexpected conditions or a flaw found in a certain unit [28]. Many faults may occur in the system such as processor faults, network faults, and physical faults. The presence of any of these faults may cause errors in the system [29]. The term "error" is usually expressed as the difference between the actual expected value and the predicted or calculated value. Error usually appears in system and application software codes which leads sometimes to certain failures in hardware components [30]. When the system fails to work properly and cannot present the expected outcome, at this point the system enters a failure state [31]. Figure 1 shows the relationship among the fault, error, and failure.

Figure 1: The Relationship Among Fault, Error, and Failure [8].

1. Fault Tolerance Parameters

For cloud computing, the current fault tolerance approach considers different parameters. The metrics are kind of fault tolerance (proactive, reactive, and resilient), performance, response time, scalability, throughput, reliability, availability, usability, overhead associated & and cost-effectiveness with it [20].

- **Performance:** Identifying the competency of the cloud service is an important step in measuring the cloud computing performance. To achieve tolerable latency, performance can be enhanced by increasing reaction time [2].
- **Response Time:** It refers to the time needed to execute the user's request in a cloud-based application. In general, the cloud providers aim to achieve minimum response time when delivering the service to the clients. The response time is usually measured in milliseconds, seconds, or minutes [34].
- **Scalability:** The increasing demands on cloud service require expanding the cloud resources. Scalability can be achieved either by adding resources such as memory, storage devices, and computing units to a certain server or by founding new data centers. This expansion enables businesses to increase the IT infrastructure for their operations [35].
- **Throughput**: It is the amount of data to be manipulated by the cloud in a certain time. This data can be a batch of tasks initiated by the cloud's users. This metric indicates the ability of the cloud to present a high level of data flow in various workload periods. Usually, throughput is measured in Mbps or MBps [35].
- **Reliability**: It refers to the ability of the cloud system to continue providing service without occurring any error. The period between every two errors in the whole cloud system is calculated, and then the mean value of these periods can be expressed as the reliability factor. It is a vital indicator that describes the system's availability [36].
- **Availability**: The probability that an object can operate properly under specific conditions at a given point in time. As a system's reliability increases, its quality is typically determined by taking into account its performance [37].
- **Cost-Effectiveness:** The fundamental aim is to access high system efficiency at a fair cost. The cost does not involve the cost of purchasing the hardware and software only, but also the ongoing costs of maintenance, support, and upgrades [38].

2. Classification of Fault Tolerance Techniques

The main scope of this review is to present a survey about the main fault tolerance techniques in cloud computing. There are three major types of fault tolerance techniques, such as reactive, proactive, and resilient methods.

2.1 Reactive Fault Tolerance Techniques

The main purpose of the reactive fault-tolerance techniques in the cloud/distributed system is to minimize the effects of the faults on the provided services to the cloud users by identifying and processing the faults after their occurrence [39]. Various reactive fault tolerance techniques are used to prevent faults in cloud computing as follows:

- **Checkpointing and Rollback:** Checkpointing is the process of periodically assigning the system state to be restored in the event of a failure. This is done by creating a snapshot of the system's resources state at a specific point in time [32]. When a failure occurs, the system can be restored to the previous checkpoint, minimizing the impact of the failure. Checkpointing is an important part of system reliability and fault tolerance. It can help to ensure that the system can recover from failures quickly and without data loss [33].
- **Replication:** It is important to periodically back up sensitive data by preparing a recovery plan in case of a robust disaster. This process can assure the organizations that data can be recovered and resume operations even if data loss or system failure [14].
- **Load Balancing:** It is the process of overcoming the overloading issue on a single server by distributing the workloads and network traffic across many. This Proactive technique ensures that a certain server cannot be overwhelmed, this will reduce the risk of performance degradation and potential failures [14].
- **Retry/ Task resubmission:** The tasks can be submitted for execution. If the task fails to be executed, then the task will be re-executing repeatedly. Re-execution may be on the same resources or another resource until the fault is repaired or it can be terminated if it is unrepairable [14].
- **S-Guard:** This technique is used in many distributed database systems which uses the rollback recovery to keep the data maintained. The main concept is to check the states of all nodes periodically. If the fault appears in a certain node, then it can be restarted from their most recent checkpoints [34].

2.2 Proactive Fault Tolerance Techniques

These techniques are responsible for addressing and identifying potential problems before causing service disruptions in the cloud system. These techniques aim to ensure the reliability of the cloud system by anticipating and preventing various faults [35]. The proactive fault tolerance techniques can be classified as the following:

- **Self-Healing:** Different VMs can execute various tasks separately. This concept makes the automatic handling of faults easier. Task automatic recovery can be performed for the isolated task, without affecting the operation of all VMs [29].
- **Machine Learning (ML) and Artificial Intelligence (AI):** ML and AI algorithms can be trained on historical data to learn patterns and anomalies, enabling them to predict potential faults. Predictive models built using ML and AI can analyze various system parameters and behavior to forecast potential faults and recommend actions to mitigate risks [29].
- **Anomaly Detection:** Anomaly detection techniques identify deviations from normal behavior or patterns in system data. Predicting faults through anomaly detection helps in identifying irregularities that may indicate potential issues, allowing for timely intervention [36].
- **Failure Mode and Effects Analysis (FMEA):** FMEA is a systematic method for evaluating and ranking potential failure modes and their effects on system performance. By systematically analyzing failure modes, FMEA helps identify critical points of failure and prioritize proactive measures to address them [13].

- **Condition Monitoring:** Overview: Continuous monitoring of the condition of system components using sensors and data analysis. Detecting early signs of degradation or abnormal conditions can help predict and prevent potential faults before they lead to system failures [37].
- **Reliability Maintenance (RCM):** RCM is a structured approach to maintenance planning that identifies critical components and optimizes maintenance strategies based on their reliability. By focusing on critical components, RCM helps predict potential failures and ensures that maintenance efforts are targeted where they are most needed [38].
- **Root Cause Analysis (RCA):** RCA is a method for identifying the root causes of faults or failures. Understanding the root causes of past failures enables organizations to predict and prevent similar issues from recurring in the future [39].
- **Failure Prediction Models:** Building models based on historical failure data to predict when similar failures may occur. These models help forecast potential faults, enabling organizations to take preventive actions, such as targeted maintenance or component replacement [40].
- **System Health Monitoring:** Continuous monitoring of various system parameters to assess overall system health. Monitoring system health allows for the early detection of signs that may indicate potential faults, enabling proactive measures to be taken [36].

D. Literature Review

Cloud providers enable their clients to access the services at any time and from anywhere based on subscription. Issues arise when resource requests are not met by the cloud environment, leaving a conundrum over how to proceed with the task. This frequent state is referred to as an equalization or resource allocation fault, and it needs to be managed without the user noticing it [41]. Numerous researchers have put out fixes for algorithms that are fault-aware and tolerance-focused.

A. Ragmania et al. analyzed the operation failure data for the heterogeneous servers, the data was collected by the Backblaze cloud storage provider between the years 2015 and 2018. The analyzing process involves investigating the feasibility of building a Machine Learning (ML)--based prediction module for cloud failure states. Many preprocessing operations have been performed to select only the 7 most affected features out of 95. Artificial Neural Networks (ANN), Naive Bayes, and regression algorithms have been utilized to build the proposed model. The results indicated that ANN achieved a better response to the failure with an accuracy of (95.55%) [42].

V. Sivaraj et al. proposed a new load allocation technique to enhance the virtualization service in cloud systems by achieving workload balancing among the VMs within the same server. This load allocation technique involved creating master and slave VMs, and then allocating different resource configurations for both. To test the proposed technique, the virtual box is used to create many VMs to be integrated for displaying the Google Earth cloud-based application in a panoramic view depending on the Liquid Galaxy framework. It is noted that integrating the different configuration VMs provided an enhanced display performance and avoided faults [43].

P. Gupta et al. Suggested and investigated a scheduling technique named "an elastic pull-based Dynamic Fault Tolerant (E-DFT)". The aim was to reduce the response time of the possible failure in the physical and virtual machine while performing backup in many independent tasks within the cloud system. E-DFT evaluation involves measuring response time, resource utilization, guarantee ratio, and energy consumption using the Cloudsim Simulation tool. The obtained results are compared with other related techniques and indicate that E-DFT achieved higher performance in terms of guarantee ratio and system utilization [44].

S. M. A. Attallah et al. proposed the "Proactive Load Balance Fault Tolerance (PLBFT)" approach to enhance the reliability and availability of cloud infrastructure. It monitors the CPU utilization, and once it detects high utilization, it moves the defective VM to another destination host or manipulates the load issue on the destination host and then moves the defective VM. The proposed approach is evaluated using Mean Time Between Failure (MTBF) and Mean Time to Fail (MTTF) metrics. The evaluation involves measuring the load balancing effectiveness with 60 faults in the VM's CPU. The results indicate that load balancing achieved higher performance in terms of operation time by moving the defective VM to a minimum load host [40].

G. Sharma presented a composition of machine learning techniques with cloud computing to build a fault tolerance framework for optimizing the system's availability and reliability. It involves multiple modules, the first module contains many nodes that collect data about CPU temperature, Memory consumption, fan speed, and response time. In addition, a prediction model based on a support vector machine algorithm is proposed to predict the failure at each node using the current and previous log data. The model was trained and tested to classify the resources' states. If there is an abnormal condition, the model classifies the data as faulty, and the alarm is turned on [45].

A. Semmoud et al. reduced the number of replications during the cloud system maintenance by designing an adaptive algorithm based on a set of previously proposed techniques for load balancing in the cloud system. The algorithm enhanced the previous techniques by mixing Migration and Replication to avoid the failure of the VMs. The evaluation involves experimenting with 200 VMs in five physical machines for sixty data centers. The evaluation focused on three parameters: overhead, the completion rate, and average CPU utilization. The proposed algorithm presented better performance compared with the other algorithms [46].

A. Rezaeipanah et al. indicated the fault detection dimension by providing a detailed analysis of the fault nature and its detection mechanism. In addition, a fuzzy logic-based method is proposed to respond to the fault occurrence by testing response time, load density, and throughput for 50 physical nodes. Optimizing fault tolerance and achieving load balancing can be requested task re-execution and migration for each checkpoint. The proposed system evaluation involves using the nodes of seven servers in the Vietnam Data Communications Company. The proposed system achieved the best accuracy compared with other related techniques [47].

J. H. Abro et al. used three machine learning algorithms to analyze the threshold values for the utilization of some cloud resources such as CPU, RAM, bandwidth, and Disk. The previous data on resource utilization was used as training and testing to assign the threshold. The data is preprocessed to extract most related features that have a direct impact on resource utilization. The Naïve Bayes, random forest, and linear regression are used to classify the data. The results indicated that Naïve Bayes can classify the data as fault or no-fault more accurately than the other algorithms in terms of accuracy, sensitivity, specificity, and mean square error. [48].

H. Yang and Y. Kim proposed a novel system to ensure cloud system availability by monitoring the fault detection process. The proposed system can be linked with a Long-Short Term Memory (LSTM) unsupervised machine learning algorithm to predict and prevent faults in the VMs. The training and testing data are collected from the real-time workload of the cloud system. In addition, the monitoring system can identify in which part of the cloud system the fault appears. After identifying the faulty part and the fault reason, the system displays the features that may cause this fault. The data have been preprocessed to be valid for training [49].

S. Jaswal and M. Malhotra designed an "Agent-based Fault Tolerance Manager (AFTM)" to overcome the failure in VMs and increase the reliability and availability of the cloud system. AFTM assigns the service provider to the cloud users based on the rank of that provider. The rank is identified based on multiple parameters such as trust values, check-pointing overheads, availability, and throughput. AFTM consists of a set of layers that interact with each other to set the rank of the service provider. The performance of the AFTM has been compared with the related OCI framework. The experimental results indicate that AFTM is better in terms of efficiency [50].

J. Gao et al. proposed a machine learning bidirectional BI-LSTM algorithm to predict the failures before their occurrence in the cloud systems and then avoid the wastage of resources. BI-LSTM analyzed the previous messaging logs of the cloud system to build a prediction model for classifying the task states as to whether it is completed or failed. The BI-LSTM model has been evaluated by comparing its performance with SVM, RNN, and LSTM. The evaluation included measuring the performance of the proposed BI-LSTM in terms of accuracy and F1 metrics for the job and task failures. The evaluation results indicated that the proposed BI-LSTM achieved higher accuracy and F1 [51].

S. Kumar T et al. focused on the checkpoint/restart method fault tolerance to improve the cloud services reliability. They proposed an "Intelligent fault-tolerant mechanism" that involved multiple algorithms and techniques. The first algorithm identifies the failure of the VM, the second enhances the interval time for each checkpoint. In the third stage, failed tasks were restarted using log-based recovery based on asynchronous checkpoint/restart. This technique was evaluated using 100 to 1000 real-time tasks within a cloud system. The results indicated the ability of the proposed technique to minimize power consumption and optimize fault tolerance compared with other related methods [52].

A. Rawat et al. suggested a "Threshold-Based Adaptive Fault Tolerance (TBAFT)" technique to support the reliability and availability of cloud services. TBAFT

detects the faulty VM and assigns the alternative according to the predefined threshold value. The adaptive manager is the main component of TBAFT, it identifies the best fault tolerance action by providing some alternative nodes within the cloud system to select the most appropriate one for the faulty node. TBAFT evaluation was compared with two fault tolerance techniques to ensure the proposed system's effectiveness. The results indicated the ability to increase fault tolerance in terms of throughput, total migration time, and failure rate [53].
P. Kumari and P. Kaur built an adaptive fault tolerance technique based on replication and checkpoint techniques. The proposed technique utilized fuzzy logic to identify the sufficient fault-tolerant method for checking the presented task in the cloud system. The identification process was performed based on many factors such as the cloud's user preferences, the failure risk of the physical machine, and the abilities of the cloud system. The proposed technique was evaluated by comparing the number of recovered hosts with checkpoints and replication separately, with the number of hosts that were recovered using the proposed adaptive systems [54].
N. R. Moparthi et al. proposed an enhanced "Energy Sensitive and Load Balancing (ESLB)" framework to manage the power and workload in the cloud-based IoT system. The proposed framework improved the IoT network response time by 60%, reducing power consumption by 31%, execution time by 24%, node shut down by 45%, and infrastructure by 48%. of the IoT network as well as minimizing the power consumption. The evaluation of the ESLB framework includes measuring the response time, execution time, and power consumption with and without the ESBL framework to identify the effectiveness of the proposed framework [55].
K. R, B et al. suggested the "Inspired Lion Optimization Algorithm for Load Balancing (ILOA-LB)" to manage the load balancing problems in the cloud system. ILOA-LB specifies the overload and underload nodes in the cloud environment, then the tasks are migrated to the appropriate nodes based on their handling capabilities. This process is performed by defining the number of nodes and identifying the fitness of each node. The task length and VM capacity are checked before migrating the task into the node. The ILOA-LB was evaluated using nodes of three data centers and the experimental results showed that ILOA-LB was able to identify the node capacity accurately [56].
S. Mangalampalli et al. proposed a "Fault Tolerance Trust-based Algorithm for Task Scheduling (FTTATS)" to enhance the cloud user's trust by avoiding task failures. The FTTATS identifies the tasks and VM priorities to schedule the task within an accurate VM. To build the scheduler, the optimization algorithm (Harris Hawks) has been utilized. FTTATS was applied to schedule tasks in many state-of-art approaches in terms of some criteria such as makespan, rate of failure, trust-based SLA, and success rate. The FTTATS was evaluated using the Cloudsim simulation tool and worklog data for real real-time supercomputer [57].
S. M. F. D. S. Mustapha and P. Gupta proposed a DBSCAN machine learning clustering algorithm to overcome resource allocation failures in cloud systems. DBSCAN can be used to divide the cloud resources nodes into many clusters based on their distances. Once all nodes have been clustered, the resources scheduler is performed in a form that all tasks within the same cluster range can be executed by

the nodes of that cluster. The evaluation involved using 1000 to 10000 tasks from four different data centers to investigate the overload and underload states. The results approved the ability of the DBSCAN to allocate the cloud resources efficiently [58].
R. R A. Muralidharan and K. Latha suggested the "Gorilla Troops Optimizer Based Fault Tolerant Aware Scheduling Scheme (GTO-FTASS)" to enhance fault tolerance during task scheduling and resource allocation. The suggested technique was inspired by the social intelligence of the gorilla troops. GTO-FTASS depends on the expected time of completion and failure probability of task execution to derive the fitness function. The proposed technique recovers the failed tasks by rescheduling them for execution again. The results showed the ability of the proposed technique to work efficiently and accurately [59].
L. Zhu et al. presented a failure recovery method based on reinforcement learning for the Automatic Train Supervision (ATS) in the cloud platform of urban rail transit. LSTM was used to enhance the fault sensitivity by classifying the service states into three classes. Then performing suitable actions such as removal or backup according to the state of each service. Three state accuracy metrics for critical, alert, and normal states are measured to evaluate the ability of an agent to monitor VMs and perform appropriate action. The experimental results indicated the ability of the LSTM to identify the ATS faults and increase the reliability [60].

E. Discussion and Comparison among Reviewed Research

In this subsection, the reviewed fault tolerance techniques are summarized and listed in Table 1. The summary involves the most related features within each reviewed work. For each reviewed work, we discussed the problem, utilized the technique, and the obtained result after applying the technique. The proposed techniques presented various solutions for a wide range of fault tolerance problems in cloud computing environments. Some of the researchers developed an existing fault tolerance technique, while others proposed a unique model as a solution.

Table 1: Comparison Among Reviewed Fault Tolerance Techniques.

Author	Problem	Category	Fault Tolerant	Technique	Result
[33]	Faults in Cloud Storage Providers	Proactive	Machine learning	ANN Algorithm	Building ANN model with 95.55%
[34]	Imbalance load for multiple VMs.	Reactive	Load Balancing	Load allocation Technique	the system is fault-free and has tolerated fault proactively.
[35]	Issues in response time issue during backup failure.	Proactive	Minimizing response time	E-DFT Scheduling technique	Minimized Response time
[25]	CPU Faults during VM operation	Reactive	Load Balancing	PLBFT	Maximum MTBF compared with some related Methods
[36]	Abnormal CPU temperature, Memory consumption, etc.	Proactive	Machine learning	SVM Algorithm	Detecting the abnormal condition of some resources.
[37]	Exhausting resources during task replication and checkpoint recovery	Reactive	Load Balancing	RPMFT	Minimizing the number of replications with reliability enhancement

[38]	Detecting fault dimension	Proactive	Condition Monitoring	Fuzzy Logic	minimizing response time to the fault occurrence
[39]	Predicting VMs failures	Proactive	Machine learning	Naïve Bayes, RF, and regression	Naïve Bayes achieved a more accurate classification
[40]	Difficulties in the configuration of the fault detection technique	Proactive	Machine learning	LSTM	
[41]	overcome the failure in VMs	Reactive	Checkpointing and Rollback	AFTM	AFTM is better in terms of efficiency
[42]	predicting job failure before the occurrence	Proactive	Machine learning	LSTM	BI-LSTM achieved higher F1 and accuracy
[43]	Enhancing checkpoint/restart method	Reactive	Checkpointing and Rollback		
[44]	Enhancing cloud service reliability	Proactive	Condition Monitoring	TBAFT	detecting and supporting the VMs in the Cloud system
[45]	Classifying the failure risks in the cloud hosts during task implementation	Reactive	Adaptive model (replication and Checkpoint)	Fuzzy Logic	Adaptive model performance is better than using checkpoint and replication separately.
[46]	Balancing the load of the IoT data and Application on the cloud	Reactive	Load Balancing	ESLB	Improving response time, execution time, power consumption, and cost.
[47]	Identifying the load capacity for cloud nodes.	Reactive	Load Balancing	Optimization Algorithms	
[48]	avoiding the task failures	Reactive	Task Scheduling	FTTATS	Enhanced performance in terms of span, failure rate, and success rate
[49]	overcome resource allocation failures in cloud system	Proactive	Machine learning	DBSCAN	DBSCAN allocated the cloud resources efficiently.
[50]	enhancing task scheduling and resource allocation	Reactive	Task Scheduling	Optimization Algorithms	Better Performance than other related techniques.
[51]	ATS faults in the urban rail transit cloud platform.	Proactive	Machine learning	LSTM	Increasing the reliability of the ATS.

In this study, the reactive and proactive categories of fault tolerance techniques are discussed. Figure 2 indicates that 50% of the reviewed techniques are reactive and 50% are proactive. The proposed reactive-based techniques implemented some procedures in cloud systems that could identify the fault and process it after occurring such as checkpoint and restarting, load balancing, and replication. The proactive-based techniques focus on monitoring the states of the VMs, services, user requests, and other system parameters to predict the possibility of a fault occurrence in any node in the system.

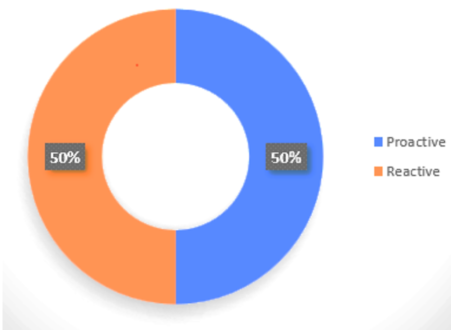

Figure 2: Categories of The Fault Tolerance Techniques Used in This Study.

The researchers built their fault tolerance frameworks to improve the reliability and availability of cloud computing services. Some of the frameworks such as [33], [36], [39], [40], [42], [49], and [51] have been designed based on machine learning algorithms as effective tools to utilize the log data of different cloud components to build classification model and predicting the future faults before occurrence. In addition, frameworks such as [25],[34],[37],[46], and [47] proposed a fault tolerance technique based on load balancing to measure the overloading level of each component to automate the load balancing among all cloud components. However, other researchers proposed their solution based on metaheuristics optimization algorithms. Figure 3 indicates the ratio of each technique reviewed in this study.

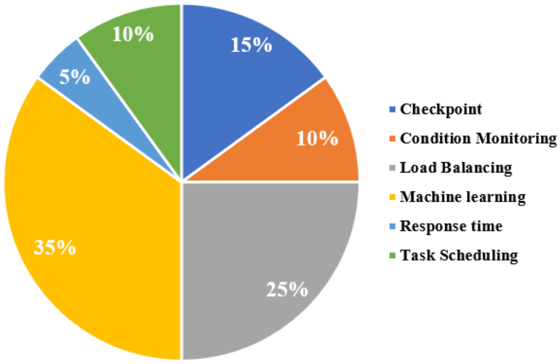

Figure 3: Summarizing the Utilized Fault Tolerance Techniques in The Study.

F. Recommendations

Based on the results in the figures (2 and 3), the process of identifying the fault is significant after and before its occurrence. It required to focus on enhancing both reactive and proactive fault tolerance techniques. In reactive techniques, it is recommended to utilize the geographical positioning system to allocate the nearest cloud servers to support the overloaded servers and achieve load balancing. In addition, KNN and K-mean machine learning algorithms can be used to divide the cloud servers into multiple clusters with homogeneous characteristics to increase the process of fault identification. In proactive techniques, it is recommended to use other machine learning and deep learning algorithms in the process of predicting the faults proactively. Optimization algorithms such as swarm and bio-inspired algorithms also could play a vital role in improving fault tolerance in cloud computing systems.

G. Conclusion

Providing cloud service with high reliability, availability, and scalability became a challenge for cloud service providers due to the complex configuration of cloud infrastructures and platforms. A variety of faults have been discussed in this study and the effects of each failure on the cloud system. Besides, multiple fault tolerance in cloud computing environments was explained. Fault tolerance is considered a crucial part of cloud system integrity, it is responsible for continuously presenting the cloud service to the cloud's users even with the presence of defects in various parts of the cloud. The main objective of this study is to present a review of the recent fault tolerance techniques. The researchers presented an effective solution to overcome the fault tolerance challenges in cloud computing. The evaluation results indicated that the proposed technique can be generalized to be utilized in another cloud environment to minimize error and optimize cloud reliability.

H. References

[1] Z. N. Rashid, S. R. M. Zeebaree, R. R. Zebari, S. H. Ahmed, H. M. Shukur, and A. Alkhayyat, "Distributed and Parallel Computing System Using Single-Client Multi-Hash Multi-Server Multi-Thread," in *2021 1st Babylon International Conference on Information Technology and Science (BICITS)*, 2021, pp. 222–227. doi: 10.1109/BICITS51482.2021.9509872.

[2] A. Lakhan *et al.*, "Secure-fault-tolerant efficient industrial internet of healthcare things framework based on digital twin federated fog-cloud networks," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 35, no. 9, p. 101747, 2023, doi: 10.1016/j.jksuci.2023.101747.

[3] S. R. M. Zeebaree, R. R. Zebari, K. Jacksi, and D. A. Hasan, "Security approaches for integrated enterprise systems performance: a review," *Int. J. Sci. Technol. Res.*, vol. 8, no. 12, pp. 2485–2489, 2019.

[4] A. H. Ibrahem and S. R. M. Zeebaree, "Tackling the Challenges of Distributed Data Management in Cloud Computing - A Review of Approaches and Solutions," *Intell. Syst. Appl. Eng.*, vol. 12, pp. 340–355, 2024.

[5] D. Cotroneo, L. De Simone, P. Liguori, and R. Natella, "Run-time failure detection via non-intrusive event analysis in a large-scale cloud computing platform," *J. Syst. Softw.*, vol. 198, p. 111611, 2023, doi: 10.1016/j.jss.2023.111611.

[6] H. S. Abdullah and S. R. M. Zeebaree, "Distributed Algorithms for Large-Scale Computing in Cloud Environments : A Review of Parallel and Distributed Processing," *Intell. Syst. Appl. Eng.*, vol. 12, pp. 356–365, 2024.

[7] K. W. Hamaali and S. R. M. Zeebaree, "Resources Allocation for Distributed Systems : A Review," *Int. J. Sci. Bus.*, vol. 5, no. 2, pp. 76–88, 2021, doi: 10.5281/zenodo.4462088.

[8] H. Bommala, U. M. V., R. Aluvalu, and S. Mudrakola, "Machine learning job failure analysis and prediction model for the cloud environment," *High-Confidence Comput.*, vol. 3, no. 4, p. 100165, 2023, doi: 10.1016/j.hcc.2023.100165.

[9] T. M. G. Sami, S. R. M. Zeebaree, and S. H. Ahmed, "Designing a New Hashing Algorithm for Enhancing IoT Devices Security and Energy Management," *Int. J. Intell. Syst. Appl. Eng.*, vol. 12, no. 4s, pp. 202–215, 2024.

[10] M. A. Mukwevho and T. Celik, "Toward a Smart Cloud: A Review of Fault-Tolerance Methods in Cloud Systems," *IEEE Trans. Serv. Comput.*, vol. 14, no. 2, pp. 589–605, Mar. 2021, doi: 10.1109/TSC.2018.2816644.

[11] H. Shukur, S. Zeebaree, R. Zebari, D. Zeebaree, O. Ahmed, and A. Salih, "Cloud Computing Virtualization of Resources Allocation for Distributed Systems," *J. Appl. Sci. Technol. Trends*, vol. 1, no. 3, pp. 98–105, 2020, doi: 10.38094/jastt1331.

[12] D. M. Abdulqader, S. R. M. Zeebaree, R. R. Zebari, S. A. Saleh, Z. N. Rashid, and M. A. M. Sadeeq, "Single-threading Based Distributed-multiprocessor-

machines Affecting by Distributed-parallel-computing Technology," *J. Duhok Univ.*, vol. 26, no. 2, pp. 416–426, 2023, doi: 10.26682/csjuod.2023.26.2.39.

[13] A. U. Rehman, R. L. Aguiar, and J. P. Barraca, "Fault-Tolerance in the Scope of Cloud Computing," *IEEE Access*, vol. 10, pp. 63422–63441, 2022, doi: 10.1109/ACCESS.2022.3182211.

[14] S. M. Alqahtani and H. Arishi, "A REVIEW ON THE TOOLS AND TECHNIQUES FOR EFFECTIVE FAILURE DETECTION AND PREDICTION IN CLOUD COMPUTING," *Manag. Appl. Sci. Technol.*, vol. 11, no. 16, pp. 11–16, 2020, doi: 10.14456/ITJEMAST.2020.315.

[15] H. Shukur, S. Zeebaree, R. Zebari, O. Ahmed, L. Haji, and D. Abdulqader, "Cache Coherence Protocols in Distributed Systems," *J. Appl. Sci. Technol. Trends*, vol. 1, no. 3, pp. 92–97, 2020, doi: 10.38094/jastt1329.

[16] Y. S. Jghef, S. R. M. Zeebaree, Z. S. Ageed, and H. M. Shukur, "Performance Measurement of Distributed Systems via Single-Host Parallel Requesting using (Single, Multi and Pool) Threads," in *2022 3rd Information Technology To Enhance e-learning and Other Application (IT-ELA)*, 2022, pp. 38–43. doi: 10.1109/IT-ELA57378.2022.10107923.

[17] T. M. G. Sami, S. R. M. Zeebaree, and S. H. Ahmed, "A Novel Multi-Level Hashing Algorithm to Enhance Internet of Things Devices' and Networks' Security," *Int. J. Intell. Syst. Appl. Eng.*, vol. 12, no. 1s, pp. 676–696, 2024.

[18] Z. S. Ageed and S. R. M. Zeebaree, "Distributed Systems Meet Cloud Computing : A Review of Convergence and Integration," *Intell. Syst. Appl. Eng.*, vol. 12, pp. 469–490, 2024.

[19] I. M. Ibrahim, S. R. M. Zeebaree, H. M. Yasin, M. A. M. Sadeeq, H. M. Shukur, and A. Alkhayyat, "Hybrid Client/Server Peer to Peer Multitier Video Streaming," in *2021 International Conference on Advanced Computer Applications (ACA)*, 2021, pp. 84–89. doi: 10.1109/ACA52198.2021.9626808.

[20] K. G. R, "Fault Tolerance in Cloud Using Reactive and Proactive Techniques," *Int. J. Comput. Sci. Eng. Commun.*, vol. 3, pp. 1159–1164, 2015, [Online]. Available: www.scientistlink.org

[21] T. M. G. Sami, S. R. M. Zeebaree, and S. H. Ahmed, "A Comprehensive Review of Hashing Algorithm Optimization for IoT Devices," *Int. J. Intell. Syst. Appl. Eng.*, vol. 11, no. 6s, pp. 205–231, 2023.

[22] M. A. M. Sadeeq and S. R. M. Zeebaree, "Design and implementation of an energy management system based on distributed IoT," *Comput. Electr. Eng.*, vol. 109, p. 108775, 2023, doi: https://doi.org/10.1016/j.compeleceng.2023.108775.

[23] Ishika and N. Mittal, "Big Data Analysis for Data VisualizationA Review," *2021 9th Int. Conf. Reliab. Infocom Technol. Optim. (Trends Futur. Dir. ICRITO 2021*, pp. 64–75, 2021, doi: 10.1109/ICRITO51393.2021.9596423.

[24] U. A. Butt *et al.*, "A review of machine learning algorithms for cloud computing security," *Electron.*, vol. 9, no. 9, pp. 1–25, 2020, doi: 10.3390/electronics9091379.

[25] A. Kadhim Gabbar Alwaeli and K. E. Kareem Al-Hamami, "Task Scheduling Algorithms in Cloud Computing," *Azerbaijan J. High Perform. Comput.*, vol. 5, no. 2, pp. 131–142, 2022, doi: 10.32010/26166127.2022.5.1.131.142.

[26] P. Kumari and P. Kaur, "A survey of fault tolerance in cloud computing,"

Journal of King Saud University - Computer and Information Sciences, vol. 33, no. 10. King Saud bin Abdulaziz University, pp. 1159–1176, Dec. 01, 2021. doi: 10.1016/j.jksuci.2018.09.021.

[27] S. M. Mohammed, K. Jacksi, and S. R. M. Zeebaree, "A state-of-the-art survey on semantic similarity for document clustering using GloVe and density-based algorithms," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 22, no. 1, pp. 552–562, 2021, doi: 10.11591/ijeecs.v22.i1.pp552-562.

[28] L. M. Haji, S. R. M. Zeebaree, Z. S. Ageed, O. M. Ahmed, M. A. M. Sadeeq, and H. M. Shukur, "Performance Monitoring and Controlling of Multicore Shared-Memory Parallel Processing Systems," in *2022 3rd Information Technology To Enhance e-learning and Other Application (IT-ELA)*, 2022, pp. 44–48. doi: 10.1109/IT-ELA57378.2022.10107953.

[29] S. Harini Krishna, G. Niveditha, and K. Gnana Mayuri, "Reliability of fault tolerance in cloud using machine learning algorithm," *Int. J. Innov. Technol. Explor. Eng.*, vol. 8, no. 12, pp. 1150–1152, Oct. 2019, doi: 10.35940/ijitee.L3889.1081219.

[30] I. M. I. Zebari, S. R. M. Zeebaree, and H. M. Yasin, "Real Time Video Streaming From Multi-Source Using Client-Server for Video Distribution," in *2019 4th Scientific International Conference Najaf (SICN)*, 2019, pp. 109–114. doi: 10.1109/SICN47020.2019.9019347.

[31] A. Ledmi, H. Bendjenna, and S. M. Hemam, "Fault Tolerance in Distributed Systems: A Survey," in *2018 3rd International Conference on Pattern Analysis and Intelligent Systems (PAIS)*, 2018, pp. 1–5. doi: 10.1109/PAIS.2018.8598484.

[32] H. Malallah *et al.*, "A Comprehensive Study of Kernel (Issues and Concepts) in Different Operating Systems," *Asian J. Res. Comput. Sci.*, no. May, pp. 16–31, 2021, doi: 10.9734/ajrcos/2021/v8i330201.

[33] N. Soveizi, F. Turkmen, and D. Karastoyanova, "Security and privacy concerns in cloud-based scientific and business workflows: A systematic review," *Futur. Gener. Comput. Syst.*, vol. 148, pp. 184–200, 2023, doi: 10.1016/j.future.2023.05.015.

[34] M. ProfMrAniljaswal, "Fault Tolerance in Cloud Computing," *Int. J. Sci. Res. Eng. Dev.*, vol. 5, 2022, [Online]. Available: www.ijsred.com

[35] S. R. M. Zeebaree, "DES encryption and decryption algorithm implementation based on FPGA," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 18, no. 2, pp. 774–781, 2020, doi: 10.11591/ijeecs.v18.i2.pp774-781.

[36] J. Liu, S. Wang, A. Zhou, S. A. P. Kumar, F. Yang, and R. Buyya, "Using proactive fault-tolerance approach to enhance cloud service reliability," *IEEE Trans. Cloud Comput.*, vol. 6, no. 4, pp. 1191–1202, Oct. 2018, doi: 10.1109/TCC.2016.2567392.

[37] A. Kumar and D. Malhotra, "Study of Various Proactive Fault Tolerance Techniques in Cloud Computing," *Int. J. Comput. Sci. Eng.*, vol. 06, no. 03, pp. 81–87, Apr. 2018, doi: 10.26438/ijcse/v6si3.8187.

[38] M. Hasan and M. S. Goraya, "Fault tolerance in cloud computing environment: A systematic survey," *Computers in Industry*, vol. 99. Elsevier B.V., pp. 156–172, Aug. 01, 2018. doi: 10.1016/j.compind.2018.03.027.

[39] D. A. Hasan, B. K. Hussan, S. R. M. Zeebaree, D. M. Ahmed, O. S. Kareem, and

M. A. M. Sadeeq, "The Impact of Test Case Generation Methods on the Software Performance: A Review," *Int. J. Sci. Bus.*, vol. 5, no. 6, pp. 33–44, 2021, doi: 10.5281/zenodo.4623940.

[40] S. M. A. Attallah, M. B. Fayek, S. M. Nassar, and E. E. Hemayed, "Proactive load balancing fault tolerance algorithm in cloud computing," *Concurr. Comput. Pract. Exp.*, vol. 33, no. 10, May 2021, doi: 10.1002/cpe.6172.

[41] P. Abdullah, H. Shukur, K. Jacksi, P. Y. Abdullah, S. R. M. Zeebaree, and H. M. Shukur, "HRM System using Cloud Computing for Small and Medium Enterprises (SMEs) Head of IT Department View project Client-Server and Video Broadcasting View project HRM System using Cloud Computing for Small and Medium Enterprises (SMEs)," *Tkru*, vol. 62, no. May, 2020, [Online]. Available: https://www.researchgate.net/publication/341883552

[42] A. Ragmani, A. Elomri, N. Abghour, K. Moussaid, M. Rida, and E. Badidi, "Adaptive fault-tolerant model for improving cloud computing performance using artificial neural network," in *Procedia Computer Science*, Elsevier B.V., 2020, pp. 929–934. doi: 10.1016/j.procs.2020.03.106.

[43] V. Sivaraj, A. Kangaiammal, and A. S. Kashyap, "Enhancing Fault Tolerance using Load Allocation Technique during Virtualization in Cloud Computing," *2021 7th Int. Conf. Adv. Comput. Commun. Syst. ICACCS 2021*, pp. 1798–1801, 2021, doi: 10.1109/ICACCS51430.2021.9441779.

[44] P. Gupta, P. K. Sahoo, and B. Veeravalli, "Dynamic fault tolerant scheduling with response time minimization for multiple failures in cloud," *J. Parallel Distrib. Comput.*, vol. 158, pp. 80–93, Dec. 2021, doi: 10.1016/j.jpdc.2021.07.019.

[45] G. Sharma, "A Practical Fault-Tolerance Approach in Cloud Computing Using Support Vector Machine," *BOHR Int. J. Smart Comput. Inf. Technol.*, vol. 2021, no. 1, pp. 14–17, 2021, doi: 10.54646/bijscit.010.

[46] A. Semmoud, M. Hakem, and B. Benmammar, "A Distributed Fault Tolerant Algorithm for Load Balancing in Cloud Computing Environments," in *E3S Web of Conferences*, EDP Sciences, May 2022. doi: 10.1051/e3sconf/202235101012.

[47] A. Rezaeipanah, M. Mojarad, and A. Fakhari, "Providing a new approach to increase fault tolerance in cloud computing using fuzzy logic," *Int. J. Comput. Appl.*, vol. 44, no. 2, pp. 139–147, 2022, doi: 10.1080/1206212X.2019.1709288.

[48] J. H. Abro *et al.*, "Artificial Intelligence Enabled Effective Fault Prediction Techniques in Cloud Computing Environment for Improving Resource Optimization," *Sci. Program.*, vol. 2022, 2022, doi: 10.1155/2022/7432949.

[49] H. Yang and Y. Kim, "Design and Implementation of Machine Learning-Based Fault Prediction System in Cloud Infrastructure," *Electron.*, vol. 11, no. 22, Nov. 2022, doi: 10.3390/electronics11223765.

[50] S. Jaswal and M. Malhotra, "AFTM-Agent Based Fault Tolerance Manager in Cloud Environment," *Int. Arab J. Inf. Technol.*, vol. 19, no. 3, pp. 396–402, 2022, doi: 10.34028/IAJIT/19/3/14.

[51] J. Gao, H. Wang, and H. Shen, "Task Failure Prediction in Cloud Data Centers Using Deep Learning," *IEEE Trans. Serv. Comput.*, vol. 15, no. 3, pp. 1411–1422, 2022, doi: 10.1109/TSC.2020.2993728.

[52] S. Kumar T, M. H S, S. M. F. D. S. Mustapha, P. Gupta, and R. P. Tripathi, “Intelligent Fault-Tolerant Mechanism for Data Centers of Cloud Infrastructure,” *Math. Probl. Eng.*, vol. 2022, 2022, doi: 10.1155/2022/2379643.

[53] A. Rawat, R. Sushil, A. Agarwal, A. Sikander, and R. S. Bhadoria, “A New Adaptive Fault Tolerant Framework in the Cloud,” *IETE J. Res.*, vol. 69, no. 5, pp. 2897–2909, 2023, doi: 10.1080/03772063.2021.1907231.

[54] P. Kumari and P. Kaur, “An Adaptable Approach to Fault Tolerance in Cloud Computing,” *Int. J. Cloud Appl. Comput.*, vol. 13, no. 1, 2023, doi: 10.4018/IJCAC.319032.

[55] N. R. Moparthi, G. Balakrishna, P. Chithaluru, M. Kolla, and M. Kumar, “An improved energy-efficient cloud-optimized load-balancing for IoT frameworks,” *Heliyon*, vol. 9, no. 11, p. e21947, 2023, doi: 10.1016/j.heliyon.2023.e21947.

[56] K. R, B. G, K. R, and V. R. R. Y, “Effective load balancing approach in cloud computing using Inspired Lion Optimization Algorithm,” *e-Prime - Adv. Electr. Eng. Electron. Energy*, vol. 6, no. October, p. 100326, 2023, doi: 10.1016/j.prime.2023.100326.

[57] S. Mangalampalli *et al.*, “Fault-Tolerant Trust-Based Task Scheduling Algorithm Using Harris Hawks Optimization in Cloud Computing,” *Sensors*, vol. 23, no. 18, Sep. 2023, doi: 10.3390/s23188009.

[58] S. M. F. D. S. Mustapha and P. Gupta, “Fault aware task scheduling in cloud using min-min and DBSCAN,” *Internet Things Cyber-Physical Syst.*, vol. 4, no. June 2023, pp. 68–76, 2024, doi: 10.1016/j.iotcps.2023.07.003.

[59] R. Rengaraj Alias Muralidharan and K. Latha, “Gorilla Troops Optimizer Based Fault Tolerant Aware Scheduling Scheme for Cloud Environment,” *Intell. Autom. Soft Comput.*, vol. 35, no. 2, pp. 1923–1937, 2023, doi: 10.32604/iasc.2023.029495.

[60] L. Zhu, Q. Zhuang, H. Jiang, H. Liang, X. Gao, and W. Wang, “Reliability-aware failure recovery for cloud computing based automatic train supervision systems in urban rail transit using deep reinforcement learning,” *J. Cloud Comput.*, vol. 12, no. 1, Dec. 2023, doi: 10.1186/s13677-023-00502-x.