A Critical Review of Faults in Cloud Computing: Types, Detection, and Mitigation Schemes

Ramandeep Kaur(✉) and V. Revathi

Dayananda Sagar University, Bangalore, India
{ramandeepkaur.res-soe-cse,revathi-cse}@dsu.edu.in

Abstract. The continuous rise in for demand services in large-scale distributed systems led to the development of cloud Computing (CC). Because it provides a combination of various software resources, CC is considered dynamically scalable. However, due to the cloud's dynamic environment, a variety of unanticipated problems and faults occur that hinder CC performance. Fault tolerance refers to a platform's capacity to respond smoothly to unanticipated hardware or programming failure. Failure must be analyzed and dealt with efficiently in cloud computing in order to accomplish high accuracy and reliability. Over the years, a significant number of techniques and approaches have been proposed for detecting the faults in CC as well as increasing their tolerance ability. In this review paper, we first provided a brief overview of Cloud computing systems, their architecture, and their working mechanism. Moreover, the services provided by Cloud computing and the issues faced by it are also highlighted in this paper. Also, the taxonomy of various faults that occur in the CC environment along with their mitigation techniques is discussed. Furthermore, it has been analyzed that traditional fault detection methods were not generating effective results which resulted in poor performance in cloud environments. Therefore, an ample number of authors stated to use Machine Learning (ML) based models for fault detection in CC. Nonetheless, ML algorithms were not able to handle a large volume of data therefore the concept of Deep Learning was introduced in fault detection approaches. Moreover, it has been also observed that the performance of DL methods can be enhanced significantly by using optimization algorithms along with them. Some of the recently proposed fault detection and tolerant systems based on ML, DL and optimization have been reviewed in this paper.

Keywords: Cloud Computing · Fault detection · Fault mitigation · Machine learning · Fault tolerance

1 Introduction

Technology name cloud computing (CC) is a modern technique that refers to browsing, modifying, and configuring software or hardware devices that are installed at far-flung locations. In the year 2009, Buyya and others [1] defined cloud computing as a type of distributed and parallel schema that involves the combination of integrated and virtualized computers that have been dynamically allocated and showcased as one or even more

S. Nandan Mohanty et al. (Eds.): ICISML 2022, LNICST 471, pp. 202–221, 2023.
https://doi.org/10.1007/978-3-031-35081-8_17

centralized computing resources. These systems generally rely on service-level agreements (SLA) that are negotiated between both the end user and the service provider. When considering the definition of Cloud computing given by the National Institute of Standards and Technology (NIST), it is defined as a strategy that enables on-demand network access to programmable computer resources. These resources include PCs, memory, connections, and programs which in return will provide immediately and deliver minimal administrative work as well as network operator involvement [2]. As cloud computing offers high reliability and on-demand services which is universally accepted in today's era. In a typical cloud system, a number of processors, storage and networking devices, and sensors are present which make it capable of processing various requests or applications received from customers. Clients send requests to the cloud system that included requests like storing data or processing apps. Every cloud-based data center is made up of physical machines (PM), including one that hosts a set of virtual machines. Such virtual machines (VMs) process the requests which are made by the client. Due to large data on cloud systems, the servers receive hundreds of requests from consumers each and every second from all over the globe to run millions of applications [3, 4].

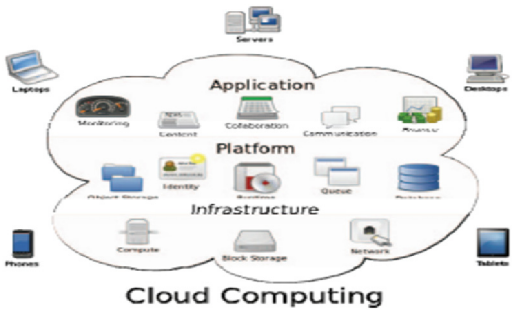

Fig. 1. Overview of Cloud computing [5]

The various services as well as an overview of cloud computing are illustrated in Fig. 1. The framework of cloud computing includes the physical layer and abstraction layer. The elements of the physical layer (including, cloud servers, storage, and various other networking components) are required to support the various facilities of the cloud. The software's present in the abstraction layer which is deployed or installed in the physical layer of the system [6] On the basis of services offered by the cloud, it can be categorized into three parts, software a service (SaaS), platform as a service (PaaS), and infrastructure as a service (IaaS). The operating system, networking devices, memory, and processor are some of the resources that the IaaS framework deals with, it also connects the client to the cloud provider. Content delivery networks (CDNs), backup and recovery, computing, and storage are some examples of IaaS. On the other hand, SaaS is an on-demand software application that allows access to different software and applications to clients (like Google docs, Email cloud, billing, customer relationship manager (CRM), financials, and so on). The cloud-like Google app engine has PaaS in it which handles the software applications across the web and permits the customer to generate applications. PaaS allows consumers to handle the complications that are faced

during managing and purchasing licenses, network development tools, and many more resources. Some examples of PaaS are business intelligence, database, development and testing, integration, and application development [7–9]. Despite the fact that cloud computing is trending among all industries, some difficulties came into existence that needs to be discussed.

1.1 Taxonomy of Faults in CC

With the ever-increasing number of web applications such as Netflix, yelp, and Pinterest installed on public cloud computing stages like Amazon EC2 to allow internet-based services, however, such web applications are prone to errors because of their dynamic, complex, and open nature of cloud systems. This in return affects a large number of customers and also causes financial losses. Among the various concerns that are faced in CC, providing continuous reliability and availability of services in a cloud system is a major challenge. Cloud systems offer services that go far beyond the traditional method which is one of the advantages of the system and these advantages have always been accompanied by some risks and difficulties. For instance, in august 2013 the biggest and most popular organization Amazon.com were unavailable for only about 45 min which results in a financial loss of around 5 million dollars [10]. In a cloud computing system, the breakdown takes place whenever there is a divergence between customer expectations and the services offered. While talking about the types of faults cloud computing is categorized into multiple domains, a few of them is related to network, media, and process-related faults. Other examples of faults in cloud computing are physical faults, service inquiry-related faults, and processor faults. The various faults and their definitions are listed below in Table 1.

Table 1. Description of various faults in cloud computing [11]

Fault types	Description
Network type	The faults that arise as a consequence of a network link failure, packet loss or packet corruption, destination failure, and many more types
Physical fault	These are the category of faults that are generally found in hardware systems which include faults in memory, faults in the CPU, and many more
Media fault	Whenever the media head collapses, these are problems that arise
Processor fault	Due to the crashes in the operating system, this type of fault takes place in the processor
Process fault	The faults that develop when there are insufficient resources, software problems, and so on
Service Expiry Faults	These are errors that happen because of the cloud service expiration for the resource selected when using the application

Besides this, based on time and resources cloud computing systems face a variety of other faults. Failures such as timing, omission, response, and crash are all forms of

failures that happen while computing on operating systems, these failures could be fixed, discontinuous, and temporary (Table 2).

Table 2. Lists of faults and their definitions in terms of Computing resources and time [12].

Fault type	Description
Permanent failures	These are the failures that arise because of natural hazards or random shutdown of power due to which several sections of the system might not operate as required
Intermittent failures	The error comes out once in a while when the operations are being carried out and it is very hard to recognize the actual harm caused by it
Transient failure	The error that appears due to an in-built fault in the system, can be rectified by repealing or rolling back the system to its earlier form. These kinds of defects are commonly detected in the computer system

Failures cause breaking/shutting down the systems, on the other hand, the distributed and cloud computing systems are distinguished by means of partial faults. The errors that take place in processing or networking elements cause a partial failure which results in the debased performance of the system rather than absolute breakdown. Although this concludes the strong and reliable systems. For a high level of performance in computing systems, the appropriate fault tolerance device is used to control the errors, even though several parts of the system are not running well this fault tolerance allows the system to assist the requirements [13, 14] (Fig. 2).

1.2 Fault Tolerance and Mitigation Techniques in CC

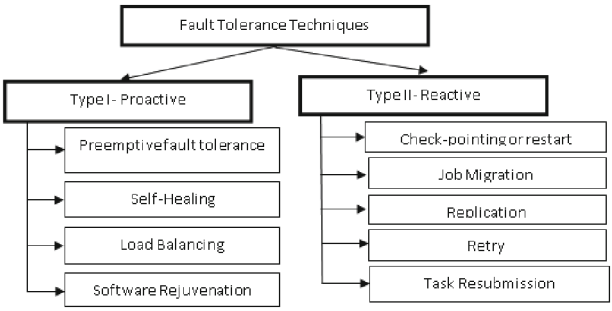

Fig. 2. Various kinds of fault tolerance Systems in Cloud Computing

Fault tolerance is a strategy or technology for designing computing systems that function in an effective manner even though some of their components are faulty. While defining a fault-tolerant system, it is stated that the system that is capable to handle

a single or even multiple faults related to different sections such as hardware-related faults, design or operation-related errors, etc., is a fault-tolerant system. [15]. The strong likelihood of faults takes place in real-time cloud systems because the processing on computing nodes is usually done distantly. Due to this reason, the requirements of the fault tolerance method are increased to obtain dependability for actual-time computing with a cloud structure. Every fault relies on one another, the connection among these faults, errors, and failures is illustrated in the below figure. The failures are caused by the faults raised in hardware or in software [16]. In a cloud computing system, there are generally 2 types of fault tolerance systems one of them is reactive and the other is termed a proactive type fault tolerance system. These systems make the overall system much stronger and more powerful to tackle faults. Figure 1, demonstrates the different fault tolerance techniques.

Proactive Fault Tolerance: The method that determines the doubtful elements and actively takes effective steps such as exchanging the suspicious content before it causes any fault. Further proactive fault tolerance systems are categorized into different types which are discussed below.

1. Preemptive Migration: Under this method, all the systems are constantly estimated and monitored; this project is based on the control loop techniques.
2. Self-Healing: These types of proactive fault tolerance discover and address all the faults that arise while executing the program and fix the failures without any delay. Along with this, a major project is divided into minor sections which operate on a virtual machine as an individual application that deals with the errors expertly.
3. Load Balancing: whenever the definite boundary is exceeded, load balancing procedures are used to equalize the burden of storage and CPU. The processing unit which exceeds its limit is being moved to that processing unit that does not surpass its maximum limit.
4. Software Rejuvenation: The technique which resets the system and makes a new beginning whenever there occurs any error in the system [2, 17].

Reactive Fault Tolerance: In this, the effect of error that arises is reduced. There are many types of reactive fault tolerance, such as.

1. Checkpointing/Restart: It works as a label, each time when the fault appears it will reboot its task to the position where it was labeled.
2. Job migration: In this method, the fault is identified for a job operating within a particular device and it will be transferred to some other machine that is suitable for that job.
3. Replication: On various devices, through separate resources, the job is copied in a broad range of times and is executed hence the requests are being fulfilled effectively.
4. Retry: similar virtual machines are required to carry out the task; the work is reduplicated numerous times till the successful outcome.
5. Task Resubmission: for the implementation, the faulted work is once more reintroduced to the same or different device [5, 12].

The Cloud framework needs to be planned in such a way that there must be the least system time off, which in turn ensures the accessibility of services to the cloud users.

To secure access to cloud services, some common policies including a replica of data and checkpointing methods are used. Fault predictions are essential for predictive maintenance because of their capability to stop breakdown events as well as maintain costs [18]. Predictive maintenance, in which the faults are expected and fetching proactive steps [19]. Great possibilities are being formed in machine learning and cloud storage to employ the data from the cloud framework which in turn will guess whenever there are errors in the component. On the basis of tools disorder, physical models, and machine learning methods, mathematical and statistical molding are the famous methods that are used for error predictions. ML-based error detecting and mitigation accesses are examined in the next part of this paper.

2 Literature Work

In order to make the CC system more efficient and reliable, it is important to detect the faults and take necessary measures like VM migration and VM load balancing, so that losses can be avoided. Conventional techniques for identifying defects were not reliable or effective due to the complicated and dynamic nature of the CC system. Various researchers have recently applied different ML techniques to forecast failure because such techniques have shown much better and good outcomes than traditional methods. Generally, a classification issue is a prediction problem that states whether there is any fault in the present cloud system or not. Future faults can be anticipated early based on historical failure patterns which in return helps to prevent losses. In this study, numerous machine learning (ML) algorithms that have recently been introduced to forecast faults in a cloud-based context are evaluated.

2.1 ML-Based Fault Detection Techniques for Fault Detection

The benefits of cloud systems always had the capability to develop maintenance issues that were increased by a huge number of clients. In current history, many scholars have designed fault tolerance mechanisms to rectify the errors in CC, but the experts were not able to handle all the issues. The authors in [20], proposed a fault detection approach, in which a fuzzy One-class support vector machine and exponentially weighted moving average method were utilized for multi-tiered web applications deployed in CC systems. In order to detect faults in the system, they used the Random Forest technique for selecting features. The suggested scheme was tested and analyzed in the private cloud where a multi-tier application was deployed by using transactional web e-commerce standard and afterward, the faults are injected into it. Experimental findings indicated the efficiency of the proposed fault detection and diagnosis approach. Moreover, the authors in [21], aim at proposing a threshold sensitivity by using support vector machines that creates an effective technique for anticipating failures in the cloud systems, in this new method the faults were avoided for every host based on log files (including CPU utilization, bandwidth, RAM and so on). When the failure threshold levels were 70%, the researchers found that this method was capable to reduce the percentage of migrations to around 76.19%. In [22] the researchers introduced a better and more effective combination based on special SVM. Initially, virtual machines were separated and classified based on their

behavioral patterns that use SVM. On the basis of behavioral patterns, by using the gossip protocol and the operation of the case for every group, QoS can be easily anticipated. The outputs of the Cloudsim simulation showed that the proposed strategy enhances processing speed by 0.65 and reduces makespan by 7.22s when compared with similar approaches. The scholars in [23] aim to lower the cost of PMs, to evaluate the resources required the modified best fit describing technique was used. Based on a fitness function, the efficiency of MBFD was optimized by using a genetic algorithm. The Polynomial Support Vector Machine (P-SVM) is used for cross-validation. That has been used for classification purposes, and parameters such as Service Level Agreement and Job Completion Ratio were evaluated. To show the efficiency of the research work a detailed comparison was made in this paper and almost 70% of improvements were noticed. The authors in [24] focused on load balancing among VMs. Certain loads such as network and CPU-based loads, and loads related to memory capacity can be handled effectively using Load balancing approaches. Such algorithm with their features assists cloud computing with multiple features such as handling failovers, improving the scalability of the system, and reducing the overall cost by providing improved resources for the process. In this study, the authors provided details that cover balancing the load using different machine learning approaches such as decision tree, SVM classifier, random forest approach, etc. The results concluded that the SVM was the approach that was handling the processing time in an effective manner by reducing it under the categorization of various user requests. The researchers evaluated and analyzed this technique in a cloud system and observed that it performed better than certain well-known methods and minimized the makespan time. A new framework namely Multi-Class Support Vector Machine was implemented for workflow scheduled in cloud systems by scholars [25] Because the NER problem was extremely hard to solve in cloud architecture. With MC-SVM, the cloud task offered an approach that generates workflow with named entity recognition. Through performing a synchronized and dynamic distribution to cloud resources, the suggested method optimized the allocation of resources mechanism and attempted to fulfill efficacy and cost objectives. The result from testing the provided technique illustrated the system's capacity to effectively manage cloud resources and minimized the makespan and cost. The authors in [26] aim at ML techniques to efficiently utilize hardware resources. Considering the present state of all virtual machines in the computer system, a prediction method named Support Vector Regression (SVR) was utilized to assess whether it is possible to redistribute resources that were planned by the genetic algorithm method. Based on the period in which the fitness function is fulfilled, the experimental results of GA and Particle Swarm were compared to evaluate which technique is better. Moreover, in the cloud architecture, the above-discussed technique provides a required output. To reduce the cloud data center's electricity usage the virtualization approach is an important technology used in CC. In this century the bulk of services was shifted to a cloud system which in turn puts more pressure on data centers, due to which the authors expanded the size of data centers. To tackle this problem, a resource allocation optimization technique was efficient as necessary. In [27] the scholars proposed a method for VM on the basis of a genetic algorithm as well as the random forest, although both the techniques were supervised ML methods. The purpose of this

work was to reduce electricity consumption while simultaneously improving load balance among available resources with help of an improved resource utilization approach. A Real-time workload trace was used to design the methodology given by Planet lab. Particularly while compared to other existing methods, the GA-RF approach improves overall energy consumption, execution time, and resource use. In [28], improved cloud reliability and preventive breakdown can be achieved by identifying and anticipating possible issues in the cloud system. Authors have undertaken numerous studies to predict potential loss, and yet no one has combined an automated system that is capable to perform tasks like monitoring, analysis, etc. by itself. Other than this few more considerations can be made such as the measurement of hard drive, and system metrics like patterns of CPU usage. The researchers offered a combined metrics strategy based on AL, as well as four artificial intelligence algorithms to review data from over a hundred cloud systems in order to detect inaccuracy. The result illustrates the benefits of integrating measures, exceeding state-of-the-art. The authors in [29] introduced a fault prediction method whitehat historical data to anticipate the failure-proneness of a component in a cloud system before that occurs. Node breakdown in systems was hard to forecast because that can be caused by a variety of aspects and expressed by a number of spatial and time signals. To resolve these concerns, the researchers proposed MING, a novel technique that integrates, the LSTM model for temporal data, a Random Forest model for spatial analysis, a ranking model that incorporated the alternate results of two modeling techniques as characteristic inputs and is currently ranked the nodes bag by their failure-proneness and a cost-sensitive function that determined the appropriate benchmark for identifying the faulty nodes. Real-world data from a cloud system were used to verify the technique and the outcomes reconfigure the effectiveness of the proposed strategy. The authors effectively implemented the proposed method in real-life situations. In [30] the experts described an unattended error detection strategy based on an ensemble of Bayesian models, which defines the system's typical operational state and recognizes abnormal activities. Once the system administrators have confirmed the irregularities, marked data was available. Afterward, to predict future cloud faults supervised learning with decision tree classifiers was utilized by the authors. According to results obtained in a research center, this method achieved an overall accuracy and even a low false positive rate for preventive fault management. The authors in [31] introduced HORA, a hierarchical online failure prediction technique that integrates component failure forecasters with architectural knowledge. To describe the fault dispersion Bayesian networks were used that included both prediction outcomes and component dependencies obtained from architecture designs. The suggested method was tested by considering 3 different types of faults that were related to memory, overload of systems, and crashing of nodes at an unexpected interval. The whole simulation was carried out by using a Netflix Server-side application that enables distributed RSS reading. By comparing the HORA with the unified method, the researchers found that the area under the ROC curve was improved by almost 9.9%. The authors in [32] investigated how resource consumption metrics from Google cluster traces could be used to detect application faults. To classify the input usage time series into different categories the recurrent neural model was used by the experts, whose results revealed that the model can anticipate batch application faults, which were the most common tasks in the Google network. However,

the scholars discovered that the forecast algorithm provides the cloud system sufficient time to take appropriate steps far sooner than the application termination which in turn results in 6–10% of resource efficiency. In [33] the researcher's aim was to identify the fundamental characteristics that were associated with cloud application faults and also provided failure forecasting models that accurately anticipated the conclusion of a task before it finished or failed. To execute this technique the authors analyzed the breakdown characteristics of the Google cluster workload trace. In order to reduce resource waste by improving system workloads that failed or are killed in the cloud system, the fault detection technique was explored more by scholars. In order to recognize application faults in the cloud, the investigators proposed a prediction approach that relies on a specialized class of recurrent neural networks called long- short-term memory. For every operation, it collects resource utilization or data performance with the purpose of forecasting the termination status. In forecasting task failures, the proposed method has an accuracy of 87%, with a true positive rate of 85% and a false positive rate of 11%. On the basis of multi-layer bidirectional long short-term memory whose purpose was to anticipate whether the projects were completed or not, the authors in [3] proposed a fault prediction approach to detect job and task errors. In trace-driven tests, this technique outperforms by 93% accuracy of previous state-of-the-art prediction algorithms an 87% accuracy for task and job failures. The comparison table for the above-mentioned fault detection techniques is given in Table 3.

Table 3. Comparison of various SVM-based fault techniques

References	Techniques Used	Results/Outcomes
[21]	Threshold sensitive by using a support vector mechanism	When the threshold = 70%, migrations were reduced to 76.19
[22]	A hybrid algorithm based on Gossip architecture by using SVM	Enhanced processing speed by 0.65 and reduces makespan by 7.22s
[23]	A genetic algorithm (GA) has been used to optimize the MBFD performance	Efficiency was improved by 70%
[24]	SVM with moth flame optimization technique	Minimized the makespan time
[25]	Multi-Class Support Vector Machine MC-SVM	Minimized the makespan and cost
[26]	Support Vector Regression (SVR), ML technique	Handled huge amounts of data on the cloud
[27]	The genetic algorithm as well as the random forest, supervised by ML methods	Reduced power usage and improved load balancing in the cloud
[28]	Failure prediction approach based on combined metrics using Artificial intelligence	Enhanced the reliability of Cloud systems

(*continued*)

Table 3. (*continued*)

References	Techniques Used	Results/Outcomes
[30]	An ensemble of Bayesian with the unsupervised learning-based fault detection method	Improved accuracy with features of low false rate
[31]	HORA, a hierarchical online failure prediction technique	Improved ROC curve by almost 9.9%
[32]	Recurrent neural networks	Achieved resource efficiency by 6–10%
[33]	LSTM model for fault detection	The overall accuracy of 87%, with a TPR and FPR of 85 and 11%respectively
[3]	A method based on multi-layer Bidirectional Long Short-Term Memory	Achieved an accuracy of 97 and 87 for task and job failure prediction

2.2 Some Other Noteworthy Works

Without virtual machines the conclusion of multiple application execution was incorrect, to resolve this issue, the authors in [34] proposed an initial virtual cluster allocation technique on the basis of the properties of VM to minimize the network resource usage as well as the energy utilized by the data center. Afterward, to predict a failed PM the CPU temperature was simulated. The investigators moved the virtual machines from degrading physical machines to a set of ideal PMs. Eventually, the best target PMs were selected as an optimization problem that was solved with an improved particle swarm optimization technique. The result demonstrated the effectiveness and efficiency of this technique. The operating metrics of all tasks executed on virtual machines were observed by authors in [35] by an agent-based monitoring method that has the capacity to fix the issues of degrading the efficiency of VM. Whenever degradation was detected, migration was started which in turn resulted in a high rate of fault tolerance and optimization techniques in terms of complexity and system performance. In [36] the researchers examined the data from the national energy research scientific computing center on the production faults over more than a five-year period. By utilizing the data from the computer failure data repository, the investigators created a successful failure forecasting model that concentrated on the high performance of the cloud system. This technique was capable of predicting future faults in the system by using an Auto-Regressive Moving Average. The result demonstrated that the failure prediction accuracy was 95%, moreover in existing real-time systems; the authors assumed that this technique was feasible and adaptable. The authors in [37] used the Google cluster workload trace to execute (which was already discussed by many other researchers in [33]) how and why problems occurred and forecast future failures. According to the research jobs that failed, consume a disproportionate number of resources, and the long short-term memory network was utilized to anticipate

the attributes that determine failure. By using the KNN classification model that was trained on the available data the termination status was forecasted. The scholars in [38] used the Enhanced-Modified Best Fit Decreasing approach to implement a VM method. Cross-validation of assigned virtual machines on real machines using Artificial Neural Networks was the conclusion of the suggested technique. Furthermore, the proposed technique seems to have the merits of recognizing false allocations that arise as a result of wasteful resource consumption and assisting in the reallocation of these VMs. In comparison to the conventional technique, the E-MBFD technique outperforms in terms of decreased power consumption as well as a lower number of service-level agreement violations, according to the experimental evidence. To forecast virtual machine breakdown by observing VM deterioration, a Fuzzy min-max Neural Network classification technique was proposed by scholars in [39]. The overloaded hyper box was categorized by using the shortest distance-based KNN classifier. FMNN was enhanced by the proposed method which results in reduced computational complexity and a quicker process. The result of framework experimentation demonstrates that VM faults can be detected ahead of time. Following the anticipation of a failure, the choice has been made to move from a problematic node to a reserve node. Thus, at a faster convergence rate, the suggested method produces more accurate forecasts. In [40], the faults in data centers and systems must have been identified and forecasted efficiently to activate the mechanism that ensures the errors that occur. When one of the hosted data centers collapses one can anticipate a failure spreading throughout the CC and execute proactive steps to deal with it. Based on information or files one approach for forecasting error was to instruct a machine to anticipate the faults among multiple elements that propagate to other data centers and exacerbate the problem. For each of the systems, parameters (including CPU consumption, RAM usage, and so on) could be maintained, and based on these elements a decision tree was developed that can evaluate whether the elements supplied into the decision tree suggested a failed state or suitable state. The authors introduced a novel methodology for forecasting VM failure based on a time series stochastic model [41]. The detection of virtual machine failure was widely expected due to several proactive fault tolerance mechanisms in cloud systems. But, accurate VM failure prediction is highly difficult and has a significant influence on the proactive fault tolerance system. Thus, a failure prediction mechanism that works on basis of time series by considering the autoregressive integrated moving average was used to predict virtual machine faults. To increase the stability of Virtual machines as well as the overall performance of cloud systems, a fault prediction approach based on AdaBoost-Hidden Markov Model was developed by researchers in [42]. The hidden Markov model was used to investigate the true relationship between the different states of the VMs such as observed and hidden. In addition to this, the impact of the AdaBoost method on the HMM and the effectiveness of forecasting VM failure was also the objective of the study. The suggested method responds to the complex and dynamic cloud system, can reliably predict VM failure state, and enhances the prediction power of VM security status, according to the results.

In [43] the authors used different strategies for anticipating failures so that users can act and repair possible faults before they occur. This technique termed PreFix aimed to detect whether a switch failure occurred in near future during runtime. The predictions were based on current switch system metrics as well as prior switch equipment failure

occurrences which are painstakingly labeled by network operators. The major finding was that the error of the same switch model shares certain similar Syslog patterns before they happen which can be extracted using ML approaches to anticipate switch failures. This innovative collection of machine learning features can properly deal with noises, sample imbalance, and computing overhead. Over the course of two years, the authors of this technique evaluated PreFix on a 9397-switch dataset under consideration of 3 different switch models. These models were installed in approximately 20-plus data centers owned by a large worldwide search engine. In PCs and ISP, prefixes outperformed the other failure prediction techniques by an average of 61.81% recall and1.84e-5 false positive ratio. Furthermore, the authors in [44] proposed a new cloud scheduling method with the goal of providing priority scheduling and fault tolerance. This approach has worked for single-server fault tolerance and reallocating problematic server activities to a new server with the least load at the time of the failure. This section also includes different strategies to implement various scheduling methods. Most of the earlier algorithm evaluations were included. In 2008, Mylara Reddy Chinnaiah, N. N. in [45] offered fault-tolerant ways describes the number of communication requests depending on the configuration raised by the various software system. These software-generated configurations were categorized into 2 types one is critical and the other non-critical. Thus, these results in adding 2 factors frequency of configuration interactions and normal interaction frequency. Both factors are helpful in measuring the system's reliability and monitoring the important software function such as financial transactions etc. It is also shown in the study that a huge count of interaction requests and failure of essential requests can cause program failure. In 2018, October-December Jialei Liu [46] introduced a proactive PCFT technique based on particle swarm optimization which was aimed at IaaS cloud architectures based on fat-tree topology. The purpose of this technique was to monitor failed PM and after that, the system looks for an optimum physical machine to migrate the VM from the failing PM. For fault detection, in the physical machine, a CPU temperature model was utilized. Based on Copy on Write- Presave in Cache (CoW-PC) Kalanirnika G R in [47] suggested a VM-Checkpoint technique (a reactive strategy) in which a VM was saved after a failure occurred, and the service was subsequently restored from the last checkpoint indicated in cache memory. To improve the memory checkpointing performance the storage exclusion approach was utilized. In [48] as a proactive fault detection technique, Zeeshan Amin April 2015 utilized the Heartbeat technique combined with an Artificial Neural Network. The technique calculated the time needed for another heartbeat signal to arrive, as well as a safety margin α. The formula used in this method is,

$$\text{Th2b} = \text{ET} + \alpha, \tag{1}$$

where Th2b is considered to be the calculation of the time interval that occurred between two heartbeat signals. In addition to this ET is termed as an estimated time that will be taken for the arrival of the next heartbeat. In [49] for cloud systems, A. Marahatta 2018 provided a fault-tolerant scheduling approach that was focusing on energy constraints. In order to predict the failure rate, researchers utilized an ML forecasting model to learn and understand the difference between both failure and non-failure-prone jobs. Afterward, the vector reconstruction strategy was employed to rebuild failure-prone activities

and projected both rectified failure-prone tasks and non-failure-prone jobs to some of the most suitable hosts. The researchers in [50] suggested an innovative and dynamic fault-tolerant scheduling strategy to help with error-free job scheduling. The provided strategy takes into account the most significant parameters such as the rate of failure and the present workload of the process-able resources for optimal QoS. Recent research showed that the proposed system was more beneficial than the baseline solutions when analyzed in terms of balancing the system load and fault tolerance. Researchers presented a fault tolerance load balancing strategy based on resource load and fault index value in [51]. The given solution was focused on 2 main functionalities the selection of resources in an effective manner and the other on the execution of a task. In the first phase, the best suitable resource was selected for implementation purposes and the resource with the lowest load and less fault occurred is selected in the task execution phase. In order to prove the effectiveness of the proposed scheme, the scheme is compared with existing state of art approaches. The model was simulated in CloudSim and proves its effectiveness in various factors such as response and makespan time, handling overhead, and providing better bandwidth. To avoid the system's real-time failure, the authors in [7] proposed a multi-level fault tolerance scheduling technique. The reliability of VMs was determined during the first step by utilizing non-functional testing and a decision-making algorithm. However, to achieve the reliability requirement, that only considered the most reliable VMs, Reliable Decision K-Nearest Neighbor was used and the scheduling mechanism was utilized to achieve high availability. For this purpose, Teaching-Learning Based Optimization scheduling method was developed, which offered a better-scheduled collection of tasks for the associated customer. The researchers evaluated the suggested method by using the Cloudsim platform. By using a multi-level structure, the result demonstrated that in a cloud context, the system provides good data availability and reliability. The researchers in [52], utilized failure statistics which were published by Backblaze (one of the leading data storage services). Between January 2015 and December 2018, the operational status of heterogeneous servers was collected in this dataset, the data was then screened and highly processed generating 2,878,440 records which include 128,820 faults. The results from the experiment indicated that the ANN model has the best prediction accuracy. In [53] the authors proposed numerous and diversified fault tolerance technique (which enables a process or an element to keep functioning normally even if a failure occur) and monitoring method (that anticipates a defect before it occurs) to increase cloud reliability. Fault tolerance techniques were important for both cloud service providers and clients and this approach contained information on the many approaches and methodologies utilized in FT, as well as a future research direction in cloud Fault tolerance. To improve cloud service reliability, different fault tolerance approaches were proposed in the literature. VM breakdown caused several challenges with cloud service dependability and availability. Moreover, by combining proactive techniques (including VM migration and duplication) and reactive methodologies (like, retry used.) the researchers in [54] contributed a new Threshold-Based Adaptive Fault Tolerance approach to cloud systems. An experimental investigation was carried out by comparing three benchmark fault tolerance methodologies (which include the first fit and least full host selection policies). By using the CloudSim simulator, the

proposed approach's study results were seen in terms of several metrics including failure rate, throughput, migration, and execution time. To minimize the user's management complexity, the authors in [55] proposed a fuzzy job distributor for fault detection administration. The suggested technique tries to decrease the probability of system faults by distributing user job requests evenly among available resources, this paper addressed abnormal conditions that could lead to failure in a self-adaptive approach by distributing incoming task requests based on the reliability of processing nodes. When compared to alternative load balancing methods, the experimental outcomes reveal a significant reduction in the occurrence of problems. The researchers in [56] presented an innovative, system-level, modular approach to establishing and maintaining fault tolerance in a Cloud system. A dedicated service layer was utilized to present a comprehensive high-level way to hide the technical details of fault tolerance approaches to application users and developers. This service layer enables the user to declare and determine the necessary level of fault tolerance while obviating a need for awareness of the fault tolerance strategies available in the anticipated Cloud system. The authors in [57], described Cluster-based, fault-tolerant mechanisms to handle the data intensiveness of scientific workflow tasks as well as data-intensive scheduling for research areas in cloud systems. A typical academic workflow was simulated, and the CFD method's outcomes were compared to three well-known heuristic scheduling policies (such as MCT, Max-min, and Min-min). When compared with the previous 3 policies, the CFD technique lowered the make-span by 14.28%, 20.37%, and 11.77%, according to the simulated results, and reduced execution costs by 1.27%, 5.3%, and 2.21%, respectively. In terms of time and cost restrictions, the SLA was not exceeded by the CFD technique, however, that was repeatedly violated by present policies. Table 4 gives a brief comparison of various above-mentioned fault detection and tolerant techniques.

Table 4. Comparison of other commonly used fault detection and tolerant techniques

References	Work done	Technique Used	Results
[34]	Proposed an initial virtual cluster allocation technique, based on VM properties	Improved Particle swarm optimization technique	Minimized the network resource use and the energy consumption by data centers
[35]	Proposed an agent-based monitoring system for VMs is developed	VM migration	Resolved the deterioration of VMs
[36]	Proposed a failure prediction model for cloud systems dataset collected from NERSC	Autoregressive moving average (ARMA)	Achieved an accuracy of 95%

(*continued*)

Table 4. (*continued*)

References	Work done	Technique Used	Results
[37]	Effectively studied the causes of failures and techniques to predict those failures by using Google cluster workload trace	LSTM and KNN	Reduced network load
[38]	Proposed a method for VM allocation	Enhanced-Modified Best Fit Decreasing (E-MBFD) Algorithm and ANN	Less power usage and SLA violations
[39]	Proposed a model for predicting the VM failures	Fuzzy min-max neural network and KNN	Achieved high accuracy with a high convergence rate
[40]	Suggested a failure predicting model by understanding the logs passed between clouds	Decision Tree	Predicted the status of the cloud system appropriately
[41]	Proposed VM fault detection method via the failure predictor	Time series-based autoregressive integrated moving average	Good failure prediction
[42]	Suggested a failure prediction model for CC	AdaBoost-Hidden Markov Model	Effectively predicted failure in virtual machines
[44]	Proposed a method for predicting single faults	Scheduling methods	Scheduled priority by using the fault-tolerant property
[45]	Suggested a fault-tolerant method that was based on communication requests and configurations	Frequency of configuration interactions (IFrFT), characteristics and frequency of interactions (ChIFrFT)	Observed various reasons for software failure
[46]	Suggested a proactive approach that directly interacts with the IaaS cloud structure	PSO and CPU temperature model	Identify and track the faulty PM
[47]	Proposed a reactive model for predicting faults in CC	VM-μ Checkpoint mechanism based on CoW-PC (Copy on Write-Presave in Cache) algorithm	Optimized the performance of checkpointing in storage

(*continued*)

Table 4. (*continued*)

References	Work done	Technique Used	Results
[48]	Proposed a proactive fault tolerance method for cloud systems	Heartbeat strategy with ANN	Predicts the time for the next heartbeat message
[49]	Developed an energy-efficient fault tolerance method	ML algorithms and vector reconstruction	The suggested model was able to predict failure effectively
[50]	Suggested yet another fault-tolerant method	Cloudsim toolkit	Balances load in CC effectively
[51]	A technique for tolerating faults was proposed	Cloudsim toolkit	Enhanced performance in terms of response time, makespan, and throughput
[7]	The fault-tolerant method was proposed	Reliable Decision K-Nearest Neighbor (RDK-NN) algorithm teaching-Learning Based Optimization (TLBO)	Tasks are scheduled more effectively
[52]	Proposed a fault detection model where they utilized Backblaze data	Self-Monitoring, Analysis, and reporting technology (SMART) along with NB, ANN	ANN is more accurate than NB in predicting faults
[54]	New Threshold-Based Adaptive Fault Tolerance (TBAFT) approach	By integrating proactive and reactive fault detection methods in Cloudsim	Reduced failure rate, high throughput
[55]	Proposed a fault tolerance technique for dealing with various failures	Fuzzy job distributor or load balancer	Minimized the fault occurrences
[57]	Proposed a CFD fault-tolerant method based on clustering	Cluster-based, fault-tolerant, and data-intensive (CFD) scheduling	CFD has less makespan time than MCT, Max-min, and min-min, reducing cost

3 Conclusion

Since the demand for cloud computing services increases day by day, therefore it is crucial to offer uninterrupted services even in presence of faults. The resources in the CC can be scaled dynamically in such a way that cost is reduced and performance is increased. The process of identifying faults and problems in a system is known as fault tolerance. The network should function properly even if a defect, hardware failure, or software failure happens. For dependable cloud computing, failures must be effectively controlled. Additionally, it will provide robustness and dependability. In this context, a

number of recently proposed fault detection and tolerance techniques are reviewed in this paper. In addition to this, we have also highlighted the basic concept of CC and various issues related to it. Furthermore, we also analyzed the role of Machine learning and DL and optimization algorithms in detecting the faults in CC and which techniques are mostly employed by researchers in their work. After analyzing the literature, we observed that a conventional number of authors have worked on VM migration, Heartbeat strategy, etc. for detecting and avoiding faults in CC. However, such methods were not able to handle faults effectivity and hence, ML algorithms were used. As there is a number of ML algorithms available, the biggest challenge for researchers was to select the best algorithm. From the literature survey, we concluded that t majority of researchers used SVM in their work for detecting and mitigating the faults in cloud systems. No doubt that ML was showing good results than traditional fault detection systems but it undergoes overfitting issues. Therefore, the authors started to use DL methods for detecting faults in cloud systems. DL-based methods, however, improved the accuracy of fault detection but it also increased latency and reduced battery life. Therefore, some researchers have used optimization-based algorithms in their work for solving multi-objective optimization problems in CC systems. it can be concluded that by integrating optimization algorithms along with effective ML or DL techniques, the accuracy of the fault detection model can be enhanced significantly.

References

1. Buyya, R., et al.: Cloud computing, and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility. Futur. Gener. Comput. Syst. **25**, 17 (2009)
2. Kumari, P., Kaur, P.: A survey of fault tolerance in cloud computing. J. King Saud Univ. Comput. Inf. Sci. **33**(10), 1159–1176 (2021)
3. Gao, J., Wang, H., Shen, H.: Task failure prediction in cloud data centers using deep learning. IEEE Trans. Serv. Comput. (2020)
4. Haresh, M., Kalady, S.: Agent-based dynamic resource allocation on federated clouds. In: Recent Advances in Intelligent Computational Systems (RAICS), pp. 111–114 (2011)
5. Ganesh, A., Sandhya, M., Shankar, S.: A study on fault tolerance methods in cloud computing. In: 2014 IEEE International Advance Computing Conference (IACC). IEEE (2014)
6. Bui, T.-K., Vo, L., Nguyen, C., Pham, T.V., Tran, H.: A fault detection and diagnosis approach for multi-tier application in cloud computing. J. Commun. Networks **22**, 399–414 (2020). https://doi.org/10.1109/JCN.2020.000023
7. Devi, K., Paulraj, D.: Multilevel fault-tolerance aware scheduling technique in cloud environment. J. Internet Technol. **22**(1), 109–119 (2021)
8. Mell, P., Grance, T.: The NIST definition of cloud computing. NIST Special Publication, 800-145 (Draft) (2011). Accessed 11 Oct 2013
9. Cloud Taxonomy. http://cloudtaxonomy.opencrowd.com
10. Wang, T., et al.: Fault detection for cloud computing systems with correlation analysis. In: 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM), pp. 652–658 (2015)
11. Prajapati, V., Thakkar, V.: A survey on failure prediction techniques in cloud computing. No. 4134. EasyChair (2020)
12. Sivagami, V.M., Easwara Kumar, K.S.: Survey on fault tolerance techniques in cloud computing environment. Int. J. Sci. Eng. Appl. Sci. **1**(9), 419–425 (2015)

13. Gokhroo, M.K., Govil, M.C., Pilli, E.S.: Detecting and mitigating faults in cloud computing environment. In: 3rd IEEE International Conference (2017)
14. Charity, T.J.: Resource reliability using fault tolerance in cloud computing, pp. 65–71 (2016)
15. Jhawar, R., Piuri, V., Santambrogio, M.: A comprehensive conceptual system-level approach to fault tolerance in cloud computing. IEEE (2012). https://doi.org/10.1109/SysCon.2012.6189503
16. Singh, A., Kinger, S.: An efficient fault tolerance mechanism based on moving averages algorithm. IJARCSSE (2013). ISSN: 2277 128X
17. Shah, Y., Thakkar, E., Bhavsar, S.: Fault tolerance in cloud and fog computing—a holistic view. In: Kotecha, K., Piuri, V., Shah, H.N., Patel, R. (eds.) Data Science and Intelligent Applications. LNDECT, vol. 52, pp. 415–422. Springer, Singapore (2021). https://doi.org/10.1007/978-981-15-4474-3_46
18. Sirbu, A., Babaoglu, O.: Towards data-driven autonomics in data centers. In: Proceedings - 2015 International Conference on Cloud Computing ICCAC 2015, pp. 45–56 (2015)
19. Pop, D.: Machine learning and cloud computing: survey of distributed and SaaS solutions. Inst. e-Austria Timisoara, Tech. Figure 2. Component Failure Analysis Rep 1 (2012)
20. Bui, K.T., et al.: A fault detection and diagnosis approach for multi-tier application in cloud computing. J. Commun. Netw. **22**(5), 399–414 (2020)
21. Fadaei Tehrani, A., Safi-Esfahani, F.: A threshold sensitive failure prediction method using support vector machine. Multiagent Grid Syst. **13**(2), 97–111 (2017)
22. Razzaghzadeh, S., Norouzi Kivi, P., Panahi, B.: A hybrid algorithm based-on Gossip architecture by using SVM for reliability in cloud computing. Soft Comput. J. (2021)
23. Singh, G., Mahajan, M.: VM Allocation in cloud computing using SVM (2019). https://doi.org/10.35940/ijitee.I1123.0789S19
24. Radhika, D., Duraipandian, M.: Load balancing in cloud computing using support vector machine and optimized dynamic task scheduling. In: 2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), pp. 1–6 (2021). https://doi.org/10.1109/ICRITO51393.2021.9596289
25. Haoxiang, W., Smys, S.: MC-SVM based workflow preparation in cloud with named entity identification. J. Soft Comput. Paradigm (JSCP) **2**(02), 130–139 (2020)
26. Anushuya, G., Gopikaa, K., Gokul Prasath, S., Keerthika, P.: Resource management in cloud computing using SVM with GA and PSO. Int. J. Eng. Res. Technol. (IJERT) ETEDM **6**(04) (2018)
27. Madhusudhan, H.S., Satish Kumar, T., Syed Mustapha, S.M.F.D., Gupta, P., Tripathi, R.P.: Hybrid approach for resource allocation in cloud infrastructure using random forest and genetic algorithm. Sci. Programm. **2021**, 10 (2021). https://doi.org/10.1155/2021/4924708
28. Chhetri, T., Dehury, C.K., Lind, A., Srirama, S.N., Fensel, A.: A combined metrics approach to cloud service reliability using artificial intelligence (2021)
29. Lin, Q., et al.: Predicting node failure in cloud systems. In: ESEC/FSE, Lake Buena Vista, FL, USA. Association for Computing Machinery (ACM) (2018)
30. Guan, Q., Zhang, Z., Fu, S.: Ensemble of bayesian predictors and decision trees for proactive failure management in cloud computing systems. J. Commun. (2012)
31. Pitakrat, T., Okanovic, D., van Hoorn, A., Grunske, L.: Hora: architecture-aware online failure prediction. J. Syst. Softw. (2018)
32. Chen, X., Lu, C., Pattabiraman, K.: Failure prediction of jobs in compute clouds: a google cluster case study. In: 2014 IEEE International Symposium on Software Reliability Engineering Workshops (2014)
33. Islam, T., Manivannan, D.: Predicting application failure in cloud: a machine learning approach. In: 2017 IEEE International Conference on Cognitive Computing (ICCC) (2017)

34. Liu, J., Wang, S., Zhou, A., Kumar, S.A.P., Yang, F., Buyya, R.: Using proactive fault-tolerance approach to enhance cloud service reliability. IEEE Trans. Cloud Comput. **6**(4), 1191–1202 (2018). https://doi.org/10.1109/TCC.2016.2567392
35. Talwar, B., Arora, A., Bharany, S.: An energy efficient agent aware proactive fault tolerance for preventing deterioration of virtual machines within cloud environment. In: 2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), pp. 1–7 (2021). https://doi.org/10.1109/ICRITO51393.2021.9596453
36. Mohammed, B., et al.: Failure analysis modelling in an infrastructure as a service (Iaas) environment. Electron. Notes Theor. Comput. Sci. **340**, 41–54 (2018)
37. Golani, B., Datta, J., Singh, G.: Prediction of cloud server job failures using machine learning based KNN classification and LSTM modelling methods. Int. J. Eng. Res. Technol. (IJERT) **10**(05) (2021)
38. Shalu, Singh, D.: Artificial neural network-based virtual machine allocation in cloud computing. J. Discrete Math. Sci. Crypt., 1–12 (2021)
39. Qasem, G.M., Madhu, B.K.: Proactive fault tolerance in cloud data centers for performance efficiency. Int. J. Pure Appl. Math. **117**(22), 325–329 (2017)
40. Bambharolia, P., Bhavsar, P., Prasad, V.: Failure prediction and detection in cloud datacenters. Int. J. Sci. Technol. Res. **6** (2017)
41. Rawat, A., Sushil, R., Agarwal, A., Afzal: A new approach for VM failure prediction using stochastic model in cloud. IETE J. Res. (2018)
42. Li, Z., Liu, L., Kong, D.: VM failure prediction method based on AdaBoost-Hidden Markov model. In: IEEE International Conference on Intelligent Transportation, Big Data and Smart City (ICITBS) (2019)
43. Zhang, S.: PreFix: switch failure prediction in datacenter networks. In: Proceedings of the ACM on Measurement and Analysis of Computing Systems (2018)
44. Singla, N., Bawa, S.: Priority scheduling algorithm with fault tolerance in cloud computing. Int. J. **3**(12) (2013)
45. Mylara Reddy Chinnaiah, N.N.: Fault-tolerant software systems using software configurations for cloud computing. J. Cloud Comput. **7** (2018). Article number: 3
46. Liu, J., Wang, S.: Using proactive fault-tolerance approach to enhance cloud service reliability. IEEE Trans. Cloud Comput. **6**(4), 1191–1202 (2018)
47. Kalanirnika, G.R., Sivagami, V.: Fault tolerance in cloud using reactive and proactive. Int. J. Comput. Sci. Eng. Commun., 1159–1164 (2015)
48. Amin, Z., Sethi, N., Singh, H.: Review of fault tolerance techniques in cloud computing. Int. J. Comput. Appl., 11–17 (2015)
49. Marahatta, C.C.: Energy-aware fault-tolerant scheduling scheme based on intelligent prediction model for cloud data center. In: 2018 Ninth International Green and Sustainable Computing Conference (IGSC), Pittsburgh, PA, USA, pp. 1–8 (2018)
50. Sathiyamoorthi, V., et al.: Adaptive fault tolerant resource allocation scheme for cloud computing environments. JOEUC **33**(5), 135–152 (2021). https://doi.org/10.4018/JOEUC.202 10901.oa7
51. Shukla, A., Kumar, S., Singh, H.: Fault tolerance based load balancing approach for web resources in cloud environment. Int. Arab J. Inf. Technol. **17**(2), 225–232 (2020)
52. Ragmani, A., et al.: Adaptive fault-tolerant model for improving cloud computing performance using artificial neural network. Procedia Comput. Sci. **170**, 929–934 (2020)
53. Gupta, A.K., Mamgain, A.: Machine learning based approach for fault tolerance in cloud computing. Int. J. Adv. Res. Ideas Innov. Technol. **4**, 59–62 (2018)
54. Rawat, A., et al.: A new adaptive fault tolerant framework in the cloud. IETE J. Res., 1–13 (2021)

55. Arabnejad, H., Pahl, C., Estrada, G., Samir, A., Fowley, F.: A fuzzy load balancer for adaptive fault tolerance management in cloud platforms. In: De Paoli, F., Schulte, S., Broch Johnsen, E. (eds.) ESOCC 2017. LNCS, vol. 10465, pp. 109–124. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-67262-5_9
56. Jhawar, R., Piuri, V., Santambrogio, M.: Fault tolerance management in cloud computing: a system-level perspective. IEEE Syst. J. **7**(2), 288–297 (2012)
57. Ahmad, Z., Jehangiri, A.I., Ala'anzy, M.A., Othman, M., Umar, A.I.: Fault-tolerant and data-intensive resource scheduling and management for scientific applications in cloud computing. Sensors (Basel) **21**(21), 7238 (2021). https://doi.org/10.3390/s21217238